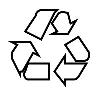


国际语言环境指南

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

部件号码: 816–3981–10
2002 年 5 月





030306@5533

内容

序 15

1 Solaris 国际化概述 21

新增的国际化和本地化特征 22

国际化和本地化的定义 23

国际化的基本步骤 23

Solaris 界面中的本地化功能 24

什么是语言环境? 25

完全语言环境和不完全语言环境 26

受语言环境影响的行为 26

语言环境种类 27

使用语言环境种类进行本地化 27

时间格式 27

日期格式 28

数字 29

货币 30

语言单词和字母差异 32

单词分隔符 32

排序顺序 32

字符集 32

键盘差异 34

纸张大小的差异 34

2 常规国际化特征 37

支持代码集独立性 37

CSI 方法 37
具有 CSI 特征 的命令 38
具有 CSI 特征 的 Solaris 9 库 39
语言环境数据库 39
进程代码格式 40
多字节支持环境 40
动态链接应用程序 40
更改的界面 41
`ctype`宏 42
`libc` 中的国际化 API 42
`genmsg` 公用程序 49
用户定义的和用户可扩展的代码转换 50

3 Solaris 9 环境中的本地化 53
本地化的软件支持 53
Solaris 9 语言环境软件包摘要 53
本版本中的附加语言环境 54
支持的语言环境 55
语言环境的多键书写序列 61
Solaris 9 产品中的键盘支持 62
在 SPARC 系统上更改键盘 63
在 Intel 系统上更改键盘 66

4 支持的亚洲语言环境 75
亚洲受支持的语言环境 75
简体和繁体中文的输入法辅助窗口支持 76
泰语本地化 77
泰语输入法辅助窗口 78
简体中文本地化 78
繁体中文本地化 81
日语本地化 84
日语语言环境 84
日语字符集 84
日语字体 84
日语输入系统 85
日语终端的终端设置 86
日语 `iconv` 模块 86

用户定义的字符支持　86
不完全语言环境和完全语言环境之间的差别　87
朝鲜语本地化　87

5　UTF-8 语言环境支持概述　91
Unicode 概述　91
Unicode 语言环境：en_US.UTF-8 支持概述　92
桌面输入法　94
语言选择和输入模式　94
访问输入法　95
繁体中文输入模式　110
繁体中文（中国香港特别行政区）输入模式　110
Unicode 十六进制输入模式　110
查表输入模式　111
系统环境　111
语言环境环境变量　111
TTY 环境设置　112
代码转换　115
DtMail　116
程序设计环境　118
用于 X 应用程序的 FontSet　119
CDE/Motif 应用程序中的 FontList 定义　119

6　复杂文本布局　121
CTL 技术概述　121
CTL 体系结构概述　122
对基于 X 库的应用程序的 CTL 支持　122
新的 XOC 资源　122
支持 CTL 技术的 Motif 更改　123
XmNlayoutDirection　123
确定布局方向　124
XmStringDirection　124
XmRendition　125
附加布局行为　125
XmText 和 XmTextField　126
字符定向操作例程　127
字符定向附加行为　127

XmText 操作例程 127
XmTextFieldGetLayoutModifier 130
XmTextGetLayoutModifier 130
XmTextFieldSetLayoutModifier 130
XmTextSetLayoutModifier 131
XmStringDirectionCreate 131
UIL Arguments 132
如何开发 CTL 应用程序 132
布局方向 132
创建绘制 133
编辑绘制 134
在资源文件中创建绘制表 135
在应用程序中创建绘制表 135
水平制表符 136
鼠标选定 137
键盘选定 138
文本资源和几何图形 138
移植说明 138

7 具有 mp 的打印过滤器增强模式 141
用于 UTF-8 的打印 141
mp 打印过滤器增强模式概述 142
与语言环境特定的字体配置文件 mp.conf 一起使用 mp 143
与语言环境特定的 PostScript 描述文件一起使用 mp 143
将 mp 用作 Xprt（X 打印服务器）客户程序 143
配置文件的本地化 143
定制现有的 prolog 文件并添加新的 prolog 文件 149
PostScript 文件定制 149
.xpr 文件定制 152
创建新的 .xpr 文件 155

A `iconv` 代码转换 **157**

B Software CD 上的不完全语言环境软件包列表 185

C Language CD 上的完整语言环境软件包列表 191

索引 **211**

表

表 1–1 国际时间格式 28
表 1–2 国际日期格式 28
表 1–3 国际数字惯例 29
表 1–4 国际货币惯例 30
表 1–5 支持欧元的用户语言环境 30
表 1–6 德国语言环境和相应的 LC_MONETARY 31
表 1–7 通用国际页大小 35
表 2–1 libc 中的信息传送函数 43
表 2–2 libc 中的代码转换 43
表 2–3 libc 中的正规表达式 43
表 2–4 libc 中的宽字符类 44
表 2–5 libc 中的修改和查询语言环境 44
表 2–6 libc 中的查询语言环境数据 44
表 2–7 libc 中的字符分类和拼写 44
表 2–8 libc 中的字符排序 46
表 2–9 libc 中的货币格式 46
表 2–10 libc 中的日期和时间格式 46
表 2–11 libc 中的多字节处理 47
表 2–12 libc 中的宽字符和字符串处理 47
表 2–13 libc 中的格式化宽字符输入和输出 48
表 2–14 宽字符串libc 49
表 2–15 libc 中的宽字符输入和输出 49
表 3–1 亚洲 55
表 3–2 澳大拉西亚 56
表 3–3 中美洲 56
表 3–4 中欧 57

表 3–5 东欧 57
表 3–6 中东 58
表 3–7 北非 58
表 3–8 北美洲 58
表 3–9 北欧 59
表 3–10 南美洲 59
表 3–11 南欧 60
表 3–12 西欧 60
表 3–13 使用书写键创建的发音字符 61
表 3–14 对地区键盘的支持 62
表 3–15 4、5 和 5c 型键盘的布局 64
表 4–1 亚洲语言环境摘要 75
表 4–2 zh_CN.EUC 语言环境的 TrueType 字体 79
表 4–3 zh_CN.EUC 语言环境的位图字体 79
表 4–4 zh_CN.GBK 语言环境的 TrueType 字体 79
表 4–5 zh_CN.GBK 语言环境的位图字体 80
表 4–6 zh_CN.GB18030 语言环境的 TrueType 字体 80
表 4–7 zh_CN.GB18030 语言环境的位图字体 80
表 4–8 简体中文代码集转换 80
表 4–9 zh_TW 语言环境的繁体中文 TrueType 字体 82
表 4–10 zh_TW 语言环境的繁体中文位图字体 82
表 4–11 zh_HK.BIG5HK 语言环境的 TrueType 字体 82
表 4–12 zh_HK.BIG5HK 语言环境的位图字体 83
表 4–13 繁体中文代码集转换 83
表 4–14 日语位图字体 85
表 4–15 日语 TrueType 字体 85
表 4–16 ko 语言环境的 Solaris 9 朝鲜语位图字体 88
表 4–17 ko.UTF-8 语言环境的 Solaris 9 朝鲜语位图字体 88
表 4–18 ko/ko.UTF-8 语言环境的 Solaris 9 朝鲜语 TrueType 字体 89
表 4–19 朝鲜语 iconv 89
表 5–1 输入模式切换键序列 96
表 5–2 常用的 Latin-1 书写键序列 97
表 5–3 通用的 Latin-2 书写键序列 101
表 5–4 通用的 Latin-3 书写键序列 103
表 5–5 通用的 Latin-5 书写键序列 104
表 5–6 通用的 Latin-9 书写键序列 104
表 5–7 en_US.UTF-8 所支持的 32 位 STREAMS 模块 112
表 5–8 en_US.UTF-8 所支持的 64 位 STREAMS 模块 112

表 6–1 XmRendition 中的新资源 125
表 6–2 Xm CTL 中的新资源 126
表 6–3 UIL 132
表 7–1 可选关键字/值对 147
表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 157
表 A–2 与 Unicode 和 IBM/Microsoft EBCDIC 及 PC 代码页相关的 iconv 可用代码转换模块 178
表 A–3 可用的 iconv 代码转换 - IBM 和 Microsoft EBCDIC/PC 代码页到 UTF-8 180
表 B–1 不完全语言环境列表 185
表 C–1 简体中文 191
表 C–2 法语 193
表 C–3 德语 195
表 C–4 意大利语 196
表 C–5 日语 198
表 C–6 朝鲜语 201
表 C–7 西班牙语 203
表 C–8 瑞典语 205
表 C–9 繁体中文 206
表 C–10 亚洲共享软件包 209

图形

图形 1–1 Solaris 操作环境中的语言环境的功能和结构 23
图形 3–1 阿拉伯语键盘 66
图形 3–2 比利时语键盘 66
图形 3–3 西里尔语（俄语）键盘 67
图形 3–4 丹麦语键盘 67
图形 3–5 芬兰语键盘 67
图形 3–6 法语键盘 68
图形 3–7 德语键盘 68
图形 3–8 意大利语键盘 68
图形 3–9 日语键盘 69
图形 3–10 朝鲜语键盘 69
图形 3–11 荷兰（荷兰语）键盘 69
图形 3–12 挪威语键盘 70
图形 3–13 葡萄牙语键盘 70
图形 3–14 西班牙语键盘 70
图形 3–15 瑞典语键盘 71
图形 3–16 瑞士（法语）键盘 71
图形 3–17 瑞士（德语）键盘 71
图形 3–18 繁体中文键盘 72
图形 3–19 土耳其语 F 型键盘 72
图形 3–20 土耳其语 Q 型键盘 72
图形 3–21 英国键盘 73
图形 3–22 美国键盘 73
图形 3–23 美国/UNIX 键盘 73
图形 4–1 辅助窗口支持的接口模型 77
图形 5–1 输入模式选择窗口 95

图形 5–2　阿拉伯文键盘　104
图形 5–3　西里尔文（俄文）键盘　105
图形 5–4　希腊文欧元键盘　105
图形 5–5　希腊文 UNIX 键盘　106
图形 5–6　希伯来文键盘　106
图形 5–7　印地文键盘　107
图形 5–8　印地文 Shift 键盘　107
图形 5–9　日语键盘　108
图形 5–10　朝鲜语键盘　108
图形 5–11　泰语键盘　109
图形 5–12　DtMail 的“新建邮件”窗口　117
图形 6–1　CTL 体系结构　122
图形 6–2　布局方向　133
图形 6–3　制表符行为　136

序

国际语言环境指南 描述了 Solaris™ 9 操作环境中新增的国际化特性。 它包含有关如何使用此版本生成支持各种语言和文化惯例的全球性软件产品的重要信息。

本出版物描述了与语言启用关联的基本属性以及由 Solaris 平台提供的有助于进行全球性应用程序开发和全世界语言服务管理的特定特性。

具体地说，本前言包含有关以下内容的信息：

- 第 15 页 "本指南的读者"
- 第 16 页 "本指南的组织"
- 第 16 页 "相关书籍和站点"

在适当的位置，本指南还为您提供了指向文档集中其它指南的链接，这些指南包含有关此版本中的国际化特性的附加或更详细的信息。 您将获得有关如何联机订购 Sun 文档以及在本指南中使用的印刷惯例的指南。

本指南的读者

本指南是供希望为 Solaris 9 操作环境设计全球性产品和应用程序的软件开发者和管理员使用的。

本指南假定您有 C 编程语言的知识。

所有操作系统信息都是关于 SunOS™ 5.9 操作环境的。

本指南的组织

本指南中的章节是以如下方式组织的：

- 第 1 章 描述 Solaris 9 中的新增国际化和本地化特性，包括在几个国家/地区中引入欧元 (€)。
- 第 2 章 描述对代码集独立性 CSI 以及 Solaris 9 产品的 libc 中的 API 的支持。
- 第 3 章 描述 Solaris 9 本地化产品的内容（包括如何本地化多语种 Solaris 产品）以及新键盘支持（包括 19 种新键盘）。
- 第 4 章 描述支持亚洲语言的语言环境、输入系统和字符支持。
- 第 5 章 探讨整合到此版本中的 en_US.UTF-8 语言环境和国际化特性。这些语言环境包括西里尔语、希腊语、 阿拉伯语、希伯来语、印地语和泰语输入法以及用于日语、朝鲜语与简体和繁体中文输入模式的输入法。
- 第 6 章 描述复杂文本布局 (CTL) 扩展，它使 Motif API 能够支持需要在逻辑和物理文字表示法之间进行复杂转换的写入系统（如阿拉伯语、希伯来语和泰语）。
- 第 7 章 解释 Solaris 9 操作环境下的打印支持，并附有与欧洲语言和亚洲语言打印以及 mp(1) 打印过滤器增强有关的特定信息。
- 附录 A 包含可用的 iconv 转换表的列表。
- 附录 B 包含 OS CD 上的不完全本地化软件包的名称表。
- 附录 C 包含表示 Language CD 上的语言软件包的表。 其中包括简体中文、法语、德语、意大利语、日语、朝鲜语、西班牙语、瑞典语、繁体中文以及共享软件包的表。

相关书籍和站点

Sun Global Application Developer Corner

有关帮助开发者对其应用程序进行全球化的信息，请参考 Sun Global Application Developer Corner (Sun GADC)

Sun Global Application Developer Corner 是以前发布的 Sun Global Application Developer Kit 1.0 的更新 Web 版本。 可以从以下站点访问它：http://www.sun.com/developers/gadc

Sun 的 Global Application Developer Corner 包含全面的国际化工具和文档，这些工具和文档用于处理在创建全球软件时遇到的各种设计和开发问题，包括如何测试全球一致性和疑难解答。

该站点包括一些测试工具（如 Sun Multibyte English (MBE) locale），它使开发者能够用伪英语测试其国际化应用程序。对于讲英语而需要测试使用特定的本地语言开发的应用程序的开发者来说，这是非常有用的。Sun Multibyte English locale 可以免费下载使用。其它有用资源包括 C 语言样例参考和代码、Solaris 操作环境中的国际语言支持白皮书、技术文章以及一些有用的供快速参考的全球化链接。其中包括一个清单，开发者可以使用它评估产品的国际化水平；还包括一个联系人页面，您可以用来就任何与 Sun 全球化相关的问题进行咨询。

Java 开发工具

有关 Java 开发工具的信息，请参见
`http://java.sun.com/j2se/1.3/docs/guide/intl/index.html`

Solaris 通用桌面环境

*Solaris Common Desktop Environment: Programmer's Guide*也是 Solaris Documentation CD 附带的 CDE 开发者集的一部分。

OSF/Motif 信息

*OSF/Motif Programmer's Guide, Release 1.2*Englewood Cliffs，New Jersey，Prentice-Hall，1993。开放软件基金会 (OSF) 指南描述了如何使用 OSF/Motif 应用程序设计接口来创建 Motif 应用程序。它提供了对 Motif 小部件集体系结构的概述，解释了 Motif 工具箱，并提供了 Motif 应用程序的模型和示例。

PostScript 信息

这套书对于进行成功的 PostScript 应用程序开发来说是必备的。

PostScript Language Reference Manual (Second Edition) 是 PostScript 的标准参考书。 它是每个 Display PostScript (DPS) 操作员（1 级和 2 级）的权威性文档。 本书介绍作为与设备无关的打印语言的 PostScript 的基础知识。并解释了 PostScript 中处理字体和字符的特殊功能。本书的附录 E 还解释了标准字符集和编码向量。它探讨了内置在解释程序内部或由其它来源提供的字体的组织。

Programming the Display PostScript System with X 是针对使用 X Windows 和 Display PostScript 的应用程序开发者编写的。 本书说明如何编写某些特定应用程序，这些程序可通过 Display PostScript 生成屏幕上显示和打印机输出的信息。它详细描述了一些编码技术。

X Window 系统已经用 X Display PostScript 系统（通常称为 X/DPS）进行了扩展。 它在客户端使用可由应用程序调用的库并在 X 服务器端使用相应的扩展。

Solaris 操作环境下的国际化进程

Tuthill、Bill 和 David Smallberg。 *Creating Worldwide Software: Solaris International Developer's Guide*，第 2 版。Mountain View，California，Sun Microsystems Press，1997。您可在 books@sun.com 以及 www.sun.com/books/ 取得这本书。本书提供对 Solaris 操作环境下的国际化进程的一般性概述。

联机访问 Sun 文档

通过 docs.sun.com[SM] Web 站点，您可以联机访问 Sun 技术文档。您可以浏览 docs.sun.com 档案或查找某一特定的书籍标题或主题。 它的 URL 为 `http://docs.sun.com`。

印刷惯例

以下表格描述了这本书中使用的印刷更改。

表 P–1 印刷惯例

字样或符号	含义	示例
`AaBbCc123`	命令名、文件名和目录名；计算机屏幕输出	编辑 `.login` 文件。 使用 `ls -a`列出所有文件。 `machine_name%` 您有邮件。
`168 AaBbCc123`	键入的内容，与计算机屏幕输出相对照	`machine_name%` **`su`** 口令：
AaBbCc123	命令行占位符： 用真实的名称或值替换	要删除文件，请键入 **`rm`** **文件名**。
AaBbCc123	书籍的标题、新词、术语或要强调的词。	请参阅《**用户指南**》第六章。 这些称为 **类** 选项。 执行此操作者，必须是**超级用户**。

命令示例中的 shell 提示符

以下表格显示了 C shell、Bourne shell 和 Korn shell 的缺省系统提示符和超级用户提示符。

表 P–2 shell 提示符

shell	提示符
C shell 提示符	machine_name%
C shell 超级用户提示符	machine_name#
Bourne shell 和 Korn shell 提示符	$
Bourne shell 和 Korn shell 超级用户提示符	#

第 1 章

Solaris 国际化概述

本部分探讨有关国际化和本地化的一些常规信息。

- 第 22 页 “新增的国际化和本地化特征”
- 第 23 页 “国际化和本地化的定义”
- 第 25 页 “什么是语言环境？”
- 第 27 页 “使用语言环境种类进行本地化”
- 第 32 页 “语言单词和字母差异”

Solaris 9 产品包含对选定的语言环境的完全 Unicode 3.1 支持（如 Unicode 和 ISO/IEC 10646 中所定义）。Solaris 9 发行版是 Sun 在国际市场的一个重要版本。它包括一些新特征。

Solaris 9 操作环境自诞生之日起便被设计为能够使用世界上的各种语言。通过可插接式、基于服务的全球化方法，Solaris 国际化体系结构简化了世界范围内的应用程序和语言服务的开发、部署和管理。在一个方便的多语种产品中，用户将能从对 39 种不同语言和 162 种语言环境（包括支持泰语和印地语所需的复杂文字布局环境以及针对阿拉伯语和希伯来语等语言的双向布局环境）的广泛支持中受益。

Solaris 国际化体系结构为处理输入法、字符集编码、代码集转换以及语言服务的其它基本方面提供了灵活的可插接式方法。您可以在所提供的功能强大的工具中进行选择，也可以定制您的环境。您可以在多个语言环境中部署应用程序，而不必了解输入法如何工作或者需要启用哪个代码集转换器，您只需遵循标准 API 即可。您还可以定制特定的语言属性。 该体系结构使您能够更改转换器表或添加新的输入法编辑器。

Solaris X 全球化框架的源代码于 2000 年秋季向公众发布。您现在能够通过遵循通用的参考实现来增强全球应用程序的兼容性和互操作性，并能同时参与基本代码的改进。独立于代码集的全球化方法使您能够在当地编码环境中进行操作或加入不断扩大的 Unicode 世界。Solaris 框架有一套功能丰富的数据转换器，用于确保各种编码和各种平台（如 Microsoft Windows 或 Macintosh）之间的互操作性，从而使您能够跨平台进行操作。

Solaris 还帮助跨国公司在世界范围内调整其服务器管理规模。与竞争性平台不同，Solaris 平台使用基于服务的方法来管理语言服务。 服务器管理员可以通过全球网络远程启用语言服务，而无论什么客户系统。通过这一与客户机无关的方法，您就可以轻松地对系统进行升级而不必更改客户机应用程序。例如，一位需要阅读从巴黎一家网吧发送的电子邮件的阿拉伯语用户仍可阅读用其本国语言撰写的电子邮件，而无须修改本地客户机应用程序。

新增的国际化和本地化特征

以下特征是 Solaris 9 发行版的新特征：

- 针对泰国、印度、中国香港特别行政区、土耳其、埃及、巴西、芬兰和比利时（瓦龙语）的附加 Unicode (UTF-8) 语言环境支持特征。
- Unicode 语言环境中的拉丁-3 字符支持。
- mp 打印过滤器中的 PCL 支持。
- 繁体中文（中国香港特别行政区）Big5+HKSCS语言环境（zh_HK.BIG5HK)。
- 繁体中文（中国香港特别行政区）UTF-8 语言环境 (zh_HK.UTF-8)。
- 泰语 UTF-8 语言环境 (th_TH.UTF-8)。
- 泰语 ISO8859–11 语言环境 (th_TH.ISO8859-11)。
- Unicode 语言环境中的印地语语言支持。
- 印地语 UTF-8 语言环境 (hi_IN.UTF-8)。
- 支持 ISCII 和 UTF-8 编码之间转换的 ISCII iconv 模块。
- 亚洲 Solaris 的排序语言环境。
- 支持新 GB18030–2000 标准的新 zh_CN.GB18030 语言环境。
- HKSCS iconv 模块。
- 新中文输入法。
- 泰语输入法增强。
- 亚洲 Solaris 的输入法辅助窗口支持。
- 针对 Fujitsu JEF、Hitachi KEIS 和 NEC JIPS 的附加日语 iconv 模块转换。
- 欧元。 只有那些使用欧元的语言环境才将欧元符号定义为它们的本国货币符号。其它 ISO8859-15 语言环境支持欧元符号 € 。
- 增强的 Unicode iconv 模块。 已针对各种新的 Unicode 编码格式以及国际和事实工业标准代码集添加并增强了 iconv 模块。
- Unicode 语言环境中的 Unicode 3.1 支持。
- 针对 ISO8859–16 的新 iconv 代码转换支持。

国际化和本地化的定义

国际化和本地化是两个不同的过程。国际化是指使软件可在多个语言或地区之间移植的过程，而本地化则是指使软件适用于特定语言或地区的过程。 国际化的软件可以使用在运行时根据特定文化要求修改程序行为的接口来开发。 本地化涉及到建立联机信息以支持某个语言或地区（称为 **语言环境**）。

与必须彻底重新编写才能用于不同当地语言和习惯的软件不同，国际化软件不需要进行重新编写。国际化的软件可在不进行更改的情况下从一个语言环境移植到另一个语言环境。 Solaris 系统是国际化的系统，提供了创建国际化软件所需的基础结构和接口。

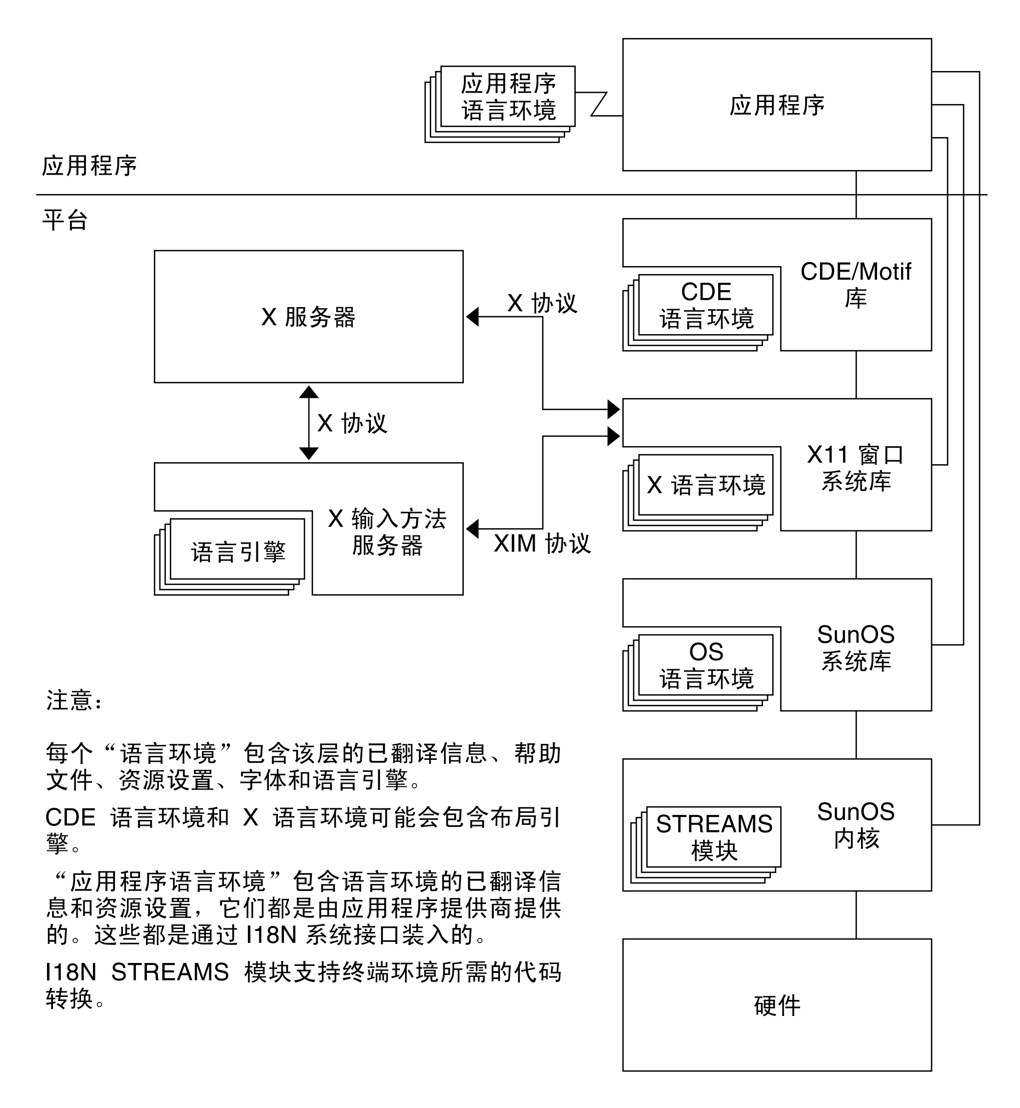
国际化的基本步骤

国际化应用程序的可执行映像可在各语言和地区之间进行移植。 要对软件进行国际化，应当：

- 使用本手册中描述的接口创建具有无须重新编译即可动态修改的环境的软件。
- 将软件分为可执行代码和信息。信息包括用户可看到的所有可打印和可显示的信息。将信息字符串保存在信息目录中。

针对某一语言或地区对信息字符串进行翻译。 **语言环境** 包括信息字符串和指定排序的方法。

要使用某个产品的本地化版本，用户需要设置一些环境变量。 然后，该产品将显示经过翻译的信息。日期、时间、货币和其它信息根据特定于语言环境的惯例进行格式设置和显示。 信息翻译和联机帮助内容在各个不同的层中提供，如下图所述。



图形 1–1 Solaris 操作环境中的语言环境的功能和结构

Solaris 界面中的本地化功能

OS 语言环境层提供基本的语言环境数据库和功能，这些数据库和功能在应用程序运行时插入到 OS 系统接口中。 应用程序通过标准 API 访问这些 OS 语言环境模块。

X11 语言环境层将该接口提供给 X 输入法和 X 输出方法，以便 X11 应用程序可以进行本地文本输入和显示。 所提供的字体使应用程序能够显示各种语言中的字符。

CDE/Motif 构建在 X11 窗口系统之上。因此，它可以通过 X11 API 使用 X11 语言环境功能。对于 CDE 应用程序，Solaris 本地化具有各种特定于语言环境的配置，从而使桌面能够在目标语言环境中工作。 在各个不同的层中提供了信息翻译和联机帮助内容。

什么是语言环境?

程序的**语言环境** 是应用程序的一个重要概念。语言环境是指当地语言环境的显式模型和定义。ANSI C 语言标准的库定义中明确定义并包含了语言环境的概念。

一个语言环境包括许多种类，对于这些种类，存在与国家/地区有关的格式设置或其它规范。一个程序的语言环境定义它的代码集、日期和时间格式设置惯例、货币惯例、十进制格式设置惯例和排序顺序。

语言环境可由基本语言、 使用国家（地区）和可选的 代码集组成。通常采用代码集。例如，德语是 `de`（Deutsch 的缩写），而瑞士德语是 `de_CH`（`CH` 是 Confederation Helvetica 的缩写）。 这使得能够区别国家/地区之间的特定差异，如货币单位表示法。

一个特定语言可以与多个语言环境相关联，这使得能够区分地区间的差异。 例如，美国的英语用户可以选择 `en_US` 语言环境（美国英语），而英国的英语用户可选择 `en_GB`（英国英语）。

语言环境名称通常由 `LANG` 环境变量指定。语言环境种类从属于 `LANG`，但也可单独设置，这时它们将覆盖 `LANG`。如果设置了 `LC_ALL` 操作数，则它不但覆盖 `LANG`，而且还将覆盖所有单独的语言环境类别。

语言环境命名惯例如下：

语言[_地区][.代码集] [@修饰符]

其中双字母**语言**代码来自 ISO 639，并且双字母**地区**代码来自 ISO 3166，**代码集**表示在该语言环境中使用的代码集的名称，而**修饰符**则表示将该语言环境与不带该修饰符的语言环境相区分的特征名称。

所有 Solaris 产品语言环境都保留了具有 US-ASCII 代码值的可移植字符集 (PCS) 字符。

有关可移植字符集的详细信息，请参考"X/Open CAE 规范： 系统接口定义，第 5 期" (ISBN 1–85912–186–1)。

一个语言环境可以有多个语言环境名称。例如 `POSIX` 与 `C` 是相同的。

完全语言环境和不完全语言环境

完全 Solaris 语言环境具有列出的所有功能和相关语言的本地化系统信息。**不完全语言环境** 未安装任何本地化信息。只要安装了相关语言的本地化信息，Solaris 环境中的所有语言环境便能够显示这些本地化信息。例如，下列语言环境既可以是不完全语言环境，也可以是完全语言环境：

- de_DE.ISO8859-1
- de_DE.ISO8859-15
- de_DE.UTF-8
- de_AT.ISO8859-1
- de_AT.ISO8859-15
- de_CH.ISO8859-1

在使用 Language CD 安装了德语信息翻译后，以上所有语言环境均成为**完全语言环境**，因为它们可以访问完全翻译了的桌面。该 Language CD 包含下列语言和语言环境的信息翻译：

- 德语
- 法语
- 西班牙语
- 瑞典语
- 意大利语
- 日语
- 朝鲜语
- 简体中文语言环境
- 繁体中文语言环境

所有不完全语言环境都可从 Software CD 中获得。 信息翻译可从 Software CD 中获得。

所有英语语言环境也是完全语言环境，并可从 Software CD 中获得。

受语言环境影响的行为

在编写日期和时间、格式化数字、分隔单词和短语以及引用材料方面，不同文化通常使用不同的惯例。 在整个系统中，语言环境决定了下列项目的行为：

- 文本数据的编码和处理。
- 标识资源文件的语言和编码。
- 文本字符串的绘制和布局。
- 交换用于客户端间文本通信的文本。
- 选择输入法（即生成哪个代码集）和文本数据的处理。
- 文化特定的字体和图标文件。
- 操作和文件类型。
- 用户界面定义 (UID) 文件。
- 日期和时间格式。
- 数字格式。

- 货币格式。
- 排序顺序。
- 特定于语言环境的正规表达式处理。
- 信息性和诊断性信息与交互式响应的格式。

Solaris 环境将语言和与文化有关的信息与应用程序分开，并在应用程序以外将其保存起来。这样，您就不必针对每个市场对应用程序进行翻译、重新编写或重新编译了。进入新市场的唯一要求就是将外部信息本地化为本地语言和习惯。

语言环境种类

语言环境种类如下所示：

LC_CTYPE	控制字符处理功能的行为。
LC_TIME	指定日期和时间格式，包括月份名称、每周日期和通用的完整表示法和缩写表示法。
LC_MONETARY	指定货币格式，包括该语言环境的货币符号、千位分隔符、符号位置、小数位数等。
LC_NUMERIC	指定十进制分隔符（或基数字符）、千位分隔符和分组。
LC_COLLATE	指定排序顺序和该语言环境的正规表达式定义。
LC_MESSAGES	指定编写本地化信息的语言，该语言环境的肯定和否定响应（yes 和 no 字符串和表达式）。
LO_LTYPE	指定提供有关语言绘制信息的布局引擎。 语言绘制（或文本绘制）由文本形状和方向性组成。

使用语言环境种类进行本地化

产品的本地化应通过向该目标语言或地区的本地用户咨询来完成。某些信息样式和格式在开发人员看来可能非常直观和常见，而在用户看来却显得很别扭或者是错误的，甚至会令人不快。以下部分描述了 Solaris 操作环境中的某些元素，您可以控制和指定这些元素以便成功地对该产品进行本地化。

时间格式

下表显示了一些不同的语言环境中 11:59 P.M 的写法。

表 1–1 国际时间格式

语言环境	格式
加拿大语	23:59
芬兰语	23.59
德语	23.59 Uhr
挪威语	23.59
泰语	23:59
大不列颠	23:59

时间用 12 小时制和 24 小时制两种方式表示。 小时和分钟分隔符可以是冒号 (:) 或句点 (.)。

在不同的国家/地区之间以及国家/地区内部存在着时区划分。 尽管时区可以根据早于或晚于协调通用时间（UTC）或格林威治时间 (GMT) 的小时数来描述，但该数字并不总是整数。例如，纽芬兰所在的时区与相邻时区相差半小时。

夏时制时间 (DST) 开始和结束的日期可能因国家/地区的不同而不同。很多国家/地区根本不实行夏时制。 此外，夏时制时间还可能在某一个时区内部存在差别。 在美国，不同的州有不同的夏时制时间。

日期格式

下表显示在世界各地使用的一些日期格式。 请注意，即使在一个国家/地区内部，日期格式可能也有所不同。

表 1–2 国际日期格式

语言环境	惯例	示例
加拿大语（英语）	dd/mm/yy	24/08/01
丹麦语	yyyy-mm-dd	2001–08–24
芬兰语	dd.mm.yyyy	24.08.2001
法语	dd/mm/yyyy	24/08/2001
德语	yyyy-mm-dd	2001–08–24
意大利语	dd/mm/yy	24/08/01
挪威语	dd-mm-yy	24–08–01
西班牙语	dd-mm-yy	24-08-01

表 1–2 国际日期格式　　（续下）

语言环境	惯例	示例
瑞典语	yyyy-mm-dd	2001-08-24
英国	dd/mm/yy	24/08/01
美国	mm-dd-yy	08-24-01
泰语	dd/mm/yyyy	24/08/2001

数字

英国和美国是世界上使用句点表示小数点位置的为数不多的几个地方中的两个。 而其它许多国家/地区都使用逗号。小数点分隔符又称为**基数** 字符。同样，英国和美国使用逗号分隔千位组，而其它许多国家/地区则使用句点，同时还有一些国家/地区使用短空格来分隔千位组。

包含特定于某一语言环境的格式的数据文件在传送到其它语言环境的系统时经常被错误地解码。例如，一个包含法语格式数字的文件不能用于特定于英国的程序。

下表显示了一些常用的数字格式。

表 1–3 国际数字惯例

语言环境	较大数字
加拿大语（英语）	4,294,967.00
丹麦语	4.294 967.295,00
芬兰语	4 294 967 295,00
法语	4 294 967 295,00
德语	4,294,967.00
意大利语	4.294.967,00
挪威语	4.294.967.295,00
西班牙语	4.294.967.295,00
瑞典语	4 294 967 295,00
英国	4,294,967,295.00
美国	4,294,967,295.00
泰语	4,294,967,295.00

注意：没有特定的语言环境惯例用来指定列表中数字的分隔方式。

货币

货币单位和表示规则在世界各地差别很大。 本地和国际货币符号可能是不同的。 下表显示了一些国家/地区的货币格式。

表 1–4 国际货币惯例

语言环境	货币	示例
加拿大语（英语）	美元 ($)	$1,234.56
加拿大语（法语）	美元 ($)	1 234,56$
丹麦语	克朗 (kr)	Kr 1.234,56
芬兰语	欧元 (€)	€ 1 234,56
法语	欧元 (€)	€ 1,234
日语	日元 (¥)	¥ 1,234
挪威语	克朗 (kr)	kr 1.234,56
瑞典语	克朗 (Kr)	1 234,56 Kr
英国	英镑 (£)	£1,234.56
美国	美元 ($)	$1,234.56
泰语	铢	2539 铢
欧元	欧元 (€)	€ 5,000

Solaris 9 软件支持欧元。 本地货币符号仍然可用，以保持向后兼容性。

表 1–5 支持欧元的用户语言环境

地区	语言环境名称	ISO 代码集
奥地利	`de_AT.ISO8859-15`	8859-15
比利时（法语）	`fr_BE.ISO8859-15`	8859-15
比利时（佛兰德语）	`nl_BE.ISO8859-15`	8859-15
丹麦	`da_DK.ISO8859-15`	8859-15
芬兰	`fi_FI.ISO8859-15`	8859-15

表 1–5 支持欧元的用户语言环境 （续下）

地区	语言环境名称	ISO 代码集
法国	fr_FR.ISO8859-15	8859-15
德国	de_DE.ISO8859-15	8859-15
爱尔兰	en_IE.ISO8859-15	8859-15
意大利	it_IT.ISO8859-15	8859-15
荷兰	nl_NL.ISO8859-15	8859-15
葡萄牙	pt_PT.ISO8859-15	8859-15
西班牙加泰罗尼亚	ca_ES.ISO8859-15	8859–15
爱沙尼亚	et_EE.ISO8859-15	8859–15
西班牙	es_ES.ISO8859-15	8859-15
瑞典	sv_SE.ISO8859-15	8859-15
英国	en_GB.ISO8859-15	8859-15
美国	en_US.ISO8859-15	8859-15

欧元语言环境基于 ISO8859–15 代码集。

请记住，**经过转换** 的货币金额可能比原始货币金额占用更多或更少的空间。 例如：$1,000 可能变为 €1.307.000。

欧元地区中语言环境的当前语言环境设置的状态由 locale 公用程序的 LC_MONETARY 操作数表示。例如，下表显示了"德国"的状态。

表 1–6 德国语言环境和相应的 LC_MONETARY

语言环境	LC_MONETARY
de_DE.ISO8859-1	DM
de_DE.ISO8859-15	Euro
de_DE.UTF-8	Euro
de_DE.ISO8859-15@euro	Euro
de_DE.UTF-8@euro	Euro

语言单词和字母差异

本部分描述各种语言之间的重要差异。

单词分隔符

在英语中，单词通常用空格字符来分隔。 而在中文、日语和泰语等语言中，单词之间通常没有分隔符。

排序顺序

特定字符的排序顺序并非在所有语言中都是相同的。例如，在德国，"ö"与普通的"o"的排序顺序相同，而在瑞典，两者却分别进行排序，其中前者位于字母表的末尾。在某些语言中，字符具有确定字符序列优先级的权值。例如，泰语词典通过重量不同的字符序列来定义排序。

字符集

各字符集在字母字符和特殊字符的数量上可能有所不同。英语字母表只包含 26 个字符，而某些语言却包含更多字符。例如，日语可包含 20000 多个字符，而中文可包含比这一数目还要多的字符。

西欧字母表

大多数西欧国家/地区的字母表类似于英语国家/地区使用的标准的 26 个字符的字母表，但往往还包含一些附加基本字符、一些带标记（或带重音）的字符和一些连字符。

日语文本

日语文本由三种不同的书写体混合而成： 源自中文的日语汉字和两种语音书写体（或音节），即平假名和片假名。

尽管平假名中的每个字符在片假名中都有一个等效字符，但平假名是最常用的书写体，具有类似草书而不是块状的字母形式。日语汉字字符用于拼写词根。片假名通常用于表示"外来"词，即从日语以外的其它语言中"引入"的词。

日语汉字包含成千上万个字符，但其中常用字符数正在逐年减少。现在经常使用的字符只有 3500 个，尽管普通日本作家的日语汉字字符的词汇量大约为 2000 个。然而，出于日本工业标准 (JIS) 的需要，计算机系统必须支持 7000 个以上的常用字符。此外，还有大约 170 个平假名和片假名字符。平均来讲，日语文本中有 55% 是平假名，35% 是日语汉字，10% 是片假名。日语文本中也包含阿拉伯数字和罗马字母。

尽管可以完全避免使用日语汉字，但大多数日语读者发现一篇不带任何日语汉字的文本很难理解。

朝鲜语文本

朝鲜语文本可使用称为“朝鲜语文字”的语音文字系统进行拼写。 朝鲜语文字包含 11000 多个字符，这些字符由称为“字母”的辅音和元音组成。 朝鲜语计算机系统中通常使用的字符大约有 3000 个，这些字符来自整个朝鲜语文字字符词汇。朝鲜语还使用名为“朝鲜语汉字”的象形字，这些象形字基于中国发明的字符集。朝鲜语文本需要 6000 多个朝鲜语汉字字符。 当朝鲜语文字表意不太明确时，常使用朝鲜语汉字来避免混淆。朝鲜语文字字符由组合在一起的辅音和元音构成。 两者组合在一起可构成音节，即朝鲜语文字字符。 朝鲜语文字字符通常排列在一个方框中，以便形成的字符组与朝鲜语汉字字符占用相同的空间。 朝鲜语文本中还包含阿拉伯数字、罗马字母和特殊符号字符。

泰语文本

一个泰语字符可在显示屏上定义为包含四个显示单元的列位置。 每个列位置最多可包含三个字符。 显示单元的组成基于泰语字符的分类。 某些泰语字符可由其它字符的分类组成。 如果可将它们组合在一起，则这两个字符将位于同一单元中。 否则，它们将分别位于不同的单元中。

中文文本

中文通常全部由称为汉字的象形书写体中的字符组成。

- 在中国 (PRC) ，GB2312（`zh` 语言环境）中大约有 7000 个常用的汉字字符，GBK 字符集（`zh.GBK` 语言环境）中有 20000 多个字符，GB18030-2000 字符集（`zh_CN.GB18030` 语言环境）中大约有 30000 个字符，包括在 Unicode 3.0 中定义的所有 CJK 扩展 A 字符。
- 在中国台湾地区，最常用的字符集是 CNS11643-1992（`zh_TW` 语言环境）和 Big5（`zh_TW.BIG5` 语言环境）。 它们共同使用大约 13000 个汉字字符。
- 在香港特别行政区，Big5 字符集中添加了 4702 个字符，从而形成 Big5-HKSCS 字符集（`zh_HK.BIG5HK`）。

如果字符不是根字符，则它通常由两个或多个部分组成，而两个部分最为常见。 在两部分字符中，一部分通常表示意义，一部分则表示发音。 有时两部分都表示意义。 偏旁部首是最重要的元素，而字符传统上是按偏旁部首进行排列的，偏旁部首的数量有几百个。一个发音可用多个不同的字符表示，这些字符是不可交替使用的。 一个字符可以有多个发音。

在给定上下文中使用某些字符比使用其它字符更为适合 — 适当的字符通过使用声调从语音上加以区分。 与此相反，日语口语和朝鲜语口语则没有声调。

表示中文的语音系统有若干种。 在中国，最常用的是**拼音**（它使用罗马字符），它在西方经常用于表示地点，如 Beijing。 威托玛式系统是一种比较陈旧的语音系统，曾经用于表示地名（如 Peking）。 而在中国台湾地区，经常使用的则是**注音**（即 **注音符号**），它是一种具有独特字母形式的语音字母表。

希伯来语文本

希伯语文本用于拼写希伯来语和意第绪语书写体，它比英语早数千年。希伯来语是一种双向书写体，这是因为希伯来语字母是从右向左进行读写的，而数字则是从左向右读的。 嵌入到希伯来语文本中的所有英语文本也是从左向右读的。

希伯来语使用 27 个字符的字母表，并使用标准拉丁语（或英语）字母表中的标点符号和数字。 希伯来语文本还包含元音和发音标记。 这些标记可以通过基本字符内部的点 (Dagesh) 、字符下的元音标记或字符左上角的重音等形式出现。 这些标记通常只用于礼拜式文本中，而很少在日常生活中见到。 希伯来语中也没有大写字母。

印地语文本

印地语文本用称为"天城书"（意思是"神的手笔"）的书写体书写。印地语是一种语音语言，并通过一系列音节进行拼写。 每个音节都由三种类型的字母片段（天城书字符）构成：辅音字母、独立元音和依赖元音符号。 音节本身由辅音和元音核心组成，同时还包含一个可选的前缀辅音。 与英语（从基准线开始）不同，天城书字符从写在字符顶部的水平线（称为头笔） 悬垂向下书写。 这些字符可根据它们的上下文进行组合或更改。 同希伯来语一样， 印地语文本不区分大小写字母。

键盘差异

并非美国键盘上的所有字符都在其它键盘中出现。 同样，其它键盘经常包含许多在美国键盘上看不到的字符。

注意：在 SPARC™ 机器上，书写键可用于生成所支持的任何 ISO8859 字符集中的任何带有音调符号的拉丁字符。

书写键可在基于拉丁文的语言环境中使用，但不能用于朝鲜语、中文或日语语言环境中（UTF-8 语言环境除外）。

由于输入由 Solaris 操作环境处理，因此可使用任何键盘输入任何语言环境中的字符。

纸张大小的差异

在每个国家/地区中，经常使用几种纸张大小。 通常情况下，其中的某个大小要比其它的大小更通用。 大多数国家/地区都遵循 ISO 标准 216：“书写纸和某些种类的打印材质-缩减大小-A 和 B 大小系列。 ”

国际化应用程序不应对可以使用的页大小进行假设。 Solaris 系统不支持对输出页大小进行跟踪。 由应用程序负责跟踪输出页大小。 下表显示了通用的国际页大小。

表 1–7 通用国际页大小

纸型	尺寸	国家/地区
ISO A4	21.0 厘米 X 29.7 厘米	除美国以外的任何地方。
ISO A5	14.8 厘米 X 21.0 厘米	除美国以外的任何地方。
JIS B4	25.9 厘米 X 36.65 厘米	日本
JIS B5	18.36 厘米 X 25.9 厘米	日本
美国信函	8.5 英寸 X 11 英寸	美国和加拿大
美国标准	8.5 英寸 X 14 英寸	美国和加拿大

第 2 章

常规国际化特征

本部分讨论 Solaris 9 环境中包含的若干个国际化特征。

- 第 37 页 “支持代码集独立性”
- 第 39 页 “语言环境数据库”
- 第 40 页 “进程代码格式”
- 第 40 页 “多字节支持环境”
- 第 40 页 “动态链接应用程序”
- 第 41 页 “更改的界面”
- 第 42 页 “`ctype`宏”
- 第 42 页 “`libc` 中的国际化 API”
- 第 49 页 “`genmsg` 公用程序”

支持代码集独立性

EUC 是扩展UNIX 码的缩写。 Solaris 9 操作环境支持非 EUC 编码，如日本的 PC-Kanji（更常用的叫法是 Shift_JIS）、中国台湾地区的 Big5 以及中华人民共和国的 GBK。 因为相当大一部分计算机市场需要非 EUC 代码集支持，所以 Solaris 9 环境提供了一个完整的框架，以同时支持 EUC 和 EUC 代码集。这种支持称为**代码集独立性** 或 CSI。

CSI 的目标是从 Solaris 操作环境库和命令中移除对特定代码集或编码方法的依赖性。CSI 体系结构使 Solaris 操作环境可以支持任何 UNIX 文件系统安全编码。CSI 支持许多新的代码集，如 UTF-8、PC-Kanji 和 Big5。

CSI 方法

代码集独立性使应用程序和平台软件开发者能够使其代码独立于任何编码（如 UTF-8），同时还提供了无须修改源代码即可采用任何新编码的能力。该体系结构方法与 Java™ 国际化的不同之处在于：Java 要求应用程序依赖于 UTF-16。

许多现有的国际化应用程序（例如，Motif）自动从基础系统继承 CSI 支持。这些应用程序无须修改即可在新的语言环境中工作。

CSI 意为本身独立于任何代码集。但是，下列关于文件代码编码（代码集）的假定依然适用于 Solaris 9 环境：

- 文件代码是 ASCII 的超集。
- 为了支持空终止多字节字符串，空字节值 (0x00) 不作为多字节字符字节的组成部分。
- 为了支持 UNIX 路径名，ASCII 斜线字符字节值 (0x2f) 不作为多字节字符字节的组成部分。

具有 CSI 特征 的命令

本部分列出了 Solaris 9 环境中具有 CSI 特征 的命令。每个命令的手册页都有一个属性部分，指明该命令是否具有 CSI 特征。

所有命令都位于 `/usr/bin` 目录中，除非另有说明。

```
/usr/lib/diffh          /usr/xpg4/bin/tr        file
/usr/sbin/accept        /usr/xpg4/bin/vedit     find
/usr/sbin/reject        /usr/xpg4/bin/vi        fold
/usr/ucb/lpr            /usr/xpg4/bin/view      ftp
/usr/xpg4/bin/awk       acctcom                 gencat
/usr/xpg4/bin/cp        apropos                 geteopt
/usr/xpg4/bin/date      batch                   getoptcvt
/usr/xpg4/bin/du        bdiff                   head
/usr/xpg4/bin/ed        cancel                  join
/usr/xpg4/bin/edit      cat                     jsh
/usr/xpg4/bin/egrep     catman                  kill
/usr/xpg4/bin/env       chgrp                   ksh
/usr/xpg4/bin/ex        chmod                   lp
/usr/xpg4/bin/expr      chown                   man
/usr/xpg4/bin/fgrep     cmp                     mkdir
/usr/xpg4/bin/lp        col                     msgfmt
/usr/xpg4/bin/ls        comm                    news
/usr/xpg4/bin/more      compress                nroff
/usr/xpg4/bin/mv        cpio                    pack
/usr/xpg4/bin/nice      csh                     paste
/usr/xpg4/bin/nohup     csplit                  pcat
/usr/xpg4/bin/od        cut                     pg
/usr/xpg4/bin/pr        diff                    printf
/usr/xpg4/bin/rm        diff3                   priocntl
/usr/xpg4/bin/sed       disable                 ps
/usr/xpg4/bin/sort      echo                    pwd
/usr/xpg4/bin/tail      expand                  rcp
```

red	split	uncompress
remsh	strconf	unexpand
rksh	strings	uniq
rsmdir	sum	unpack
rsh	tabs	wc
script	tar	whatis
sdiff	tee	write
settime	touch	xargs
sh	tty	zcat

具有 CSI 特征 的 Solaris 9 库

libc (/usr/lib/libc.so) 中的几乎所有函数都具有 CSI 特征。但是，libc 中的下列函数不具有 CSI 特征，因为它们是依赖于 EUC 的函数：

- csetcol()
- csetlen()
- euccol()
- euclen()
- eucscol()
- getwidth()
- csetno()
- wcsetno()

在 Solaris 9 产品中，libgen /usr/ccs/lib/libgen.a 和 libcurses /usr/ccs/lib/libcurses.a 被国际化但不具有 CSI 特征。

语言环境数据库

语言环境数据库格式和结构是专用的，并且在将来的发行版中可能被修改。因而，在开发国际化的应用程序时，不要直接访问语言环境数据库。相反，请使用 libc 中的国际化 API，如第 42 页"libc 中的国际化 API" 中所述。

注意：在使用 Solaris 9 环境时，请使用 Solaris 9 产品随附的语言环境数据库。不要使用以前的 Solaris 版本中的语言环境。

进程代码格式

进程代码格式（在 Solaris 9 产品中，也称为宽字符代码格式）是专用的，并且在将来的发行版中可能被修改。因而，在开发国际应用程序时，不要假定进程代码格式是相同的。相反，请使用 `libc` 中的国际化 API，如 第 42 页“`libc` 中的国际化 API” 中所述。

注意：所有 Unicode 语言环境的进程代码都在 UTF-32 表示法中。有关 UTF-32 的详细信息，请参考“Unicode 标准附件 19 号： UTF 32”和“Unicode 标准附件 27 号：Unicode 3.1”（它们可以从 Unicode 协会或 http://www.unicode.org/ 得到）。

多字节支持环境

多字节字符是无法存储在单个字节中的字符，如中文、日语或韩文字符。这些字符需要 2、3 或 4 个字节的存储空间。可在 ISO/IEC 9899:1990 3.13 款中找到更准确的定义。

ANSI C 修订 1（也称为 ISO/IEC 9899:1990）添加了新的国际化特征，一并称为多字节支持环境 (MSE)。修订 1 为具有状态的多字节代码集定义了附加的国际化 API，这同时也是为了更好地支持宽字符处理。

程序设计模型使这些多字节字符能够作为逻辑单元读入，然后在内部存储为宽字符。这些宽字符可由程序作为具有自身权限的逻辑实体进行处理。最后，这些宽字符可在经过适当的转换后作为逻辑单元写出。

此过程类似于将单字节字符读入、进行处理再写出的方式。MSE 使人们能够使用与处理单字节字符相同的程序设计模型来编写处理多字节字符的程序。

动态链接应用程序

Solaris 9 产品用户可以选择如何用系统库（如 `libc`）来链接应用程序，可以使用动态链接，也可以使用静态链接。任何需要系统库中的国际化特征的应用程序都必须动态链接。如果应用程序是被静态链接的，则在使用 `setlocale` 函数将语言环境设置为除 C 和 POSIX 以外的语言环境时，操作将失败。静态链接的应用程序只能在 C 和 POSIX 语言环境中运行。

默认情况下，链接程序尝试动态链接应用程序。如果链接程序和编译程序的命令行选项包括 -Bstatic 或 -dn 指定内容，则应用程序可能被静态链接。可以使用 /usr/bin/ldd 命令检查现有的应用程序是否为动态链接的。

例如，如果键入：

```
% /usr/bin/ldd /sbin/sh
```

该命令指明 /sbin/sh 命令不是动态链接程序，如下面的响应所示：

```
ldd: /sbin/sh: file is not a dynamic executable or shared object
```

如果您键入：

```
% /usr/bin/ldd /usr/bin/ls
```

该命令将显示下列信息：

```
libc.so.1 =>     /usr/lib/libc.so.1
libdl.so.1 => /usr/lib/libdl.so.1
```

此信息表明 /usr/bin/ls 命令已经与以下两个库动态链接：libc.so.1 和 libdl.so.1。

更改的界面

libw 和 libintl 已经移到 libc 中，而不再位于 libw 和 libintl中。

共享对象确保现有应用程序的运行兼容性，并且和归档文件一起为生成应用程序提供了编译环境兼容性。然而，您不再必须针对 libw 或 libintl 生成应用程序。

有关过滤器的更多信息，请参见《*Linker and Libraries Guide*》。

以下列表说明了 libw 中的桩模块入口点。

fgetwc	iswalnum	putwc	wcscpy
fgetws	iswalpha	putwchar	wcscspn
fputwc	iswcntrl	putws	wcsftime
fputws	iswctype	strtows	wcsclen
getwc	iswdigit	towlower	wscncat
getwchar	iswgraph	towupper	wcsncmp
getws	iswlower	ungetwc	wcsncpy
isenglish	iswprint	watoll	wcspbrk
isideogram	iswpunct	wcscat	wcsrchr
isnumber	iswspace	wcschr	wcsspn
isphonogram	iswupper	wcscmp	wcstod
isspecial	iswxdigit	wcscoll	wcstok

wcstol	wscat	wslen	wsscanf
wcstoul	wschr	wsncasecmp	wsspn
wcswcs	wscmp	wsncat	wstod
wcswidth	wscol	wsncmp	wstok
wcsxfrm	wscoll	wsncpy	wstol
wctype	wscpy	wspbrk	wstoll
wcwidth	wscspn	wsprintf	wstostr
wscasecmp	wsdup	wsrchr	wsxfrm

下面这个较短的列表包含 libintl 中的桩模块入口点：

```
bindtextdomain
dcgettext
dgettext
gettext
textdomain
```

ctype宏

字符 分类和字符变换宏在 /usr/include/ctype.h 中定义。 Solaris 9 环境提供了一组ctype支持 XPG4 所定义的字符分类和变换语义的宏。为了使所有 XPG4 和 XPG4.2 应用程序自动访问新的宏，必须满足下列条件之一：

- 定义了 _XPG4_CHAR_CLASS。
- 定义了 _XOPEN_SOURCE 和 _XOPEN_VERSION=4 。
- 定义了 _XOPEN_SOURCE 和 _XOPEN_SOURCE_EXTENDED=1 。

因为 _XOPEN_SOURCE、_XOPEN_VERSION 和 _XOPEN_SOURCE_EXTENDED 除了引入新的 ctype 宏以外，还引入了附加的 XPG4 相关特征，所以非 XPG4 或 XPG4.2 应用程序应该使用 __XPG4_CHAR_CLASS__。

还存在相应的 ctype 函数。Solaris 9 环境函数也支持 XPG4 语义。有关详细信息，请参考 ctype(3C) 手册页。

libc 中的国际化 API

Solaris 9 环境提供两组 API：

- 多字节（文件代码）
- 宽字符（进程代码）

宽字符代码是固定宽度的逻辑实体单元。因此，在使用多字节字符时，不必跟踪维护正确的字符边界。

当程序从文件获取输入时，可以将文件的多字节数据转换为宽字符进程代码，方法是直接使用 fscanf(3S) 和 fwscanf(3S) 等输入函数，或者在输入之后使用 mbtowc(3C) 和 mbsrtowcs(3C) 等转换函数。要将输出数据从宽字符格式转换为多字节字符格式，请使用 fwprintf(3S) 和 fprintf(3S) 等输出函数，或者在输出之前应用 wctomb(3C) 和 wcsrtombs(3C) 等转换函数。

本章剩余部分中的表格描述了 Solaris 9 产品中包含的国际化 API。

下表描述了 libc 中的信息传送函数 API。

表 2–1 libc 中的信息传送函数

库例程	说明
catclose()	关闭信息目录
catgets()	读取程序信息
catopen()	打开信息目录
dgettext()	指定了域，从信息目录中获取信息
dcgettext()	指定了域和种类，从信息目录中获取信息
textdomain()	设置并查询当前域
bindtextdomain()	绑定信息域的路径
gettext()	从信息数据库中检索文本字符串

下表描述了 libc 中的代码转换函数 API。

表 2–2 libc 中的代码转换

库例程	说明
iconv()	转换代码
iconv_close()	解除分配转换描述符
iconv_open()	分配转换描述符

下表描述了 libc 中的正规表达式 API。

表 2–3 libc 中的正规表达式

库例程	说明
regcomp()	编译正规表达式

表 2–3 libc 中的正规表达式 （续下）

库例程	说明
regexec()	执行正规表达式匹配
regerror()	提供从错误代码到错误信息的映射
regfree()	释放由 regcomp() 分配的内存
fnmatch()	匹配文件名或路径名

下表描述了 libc 中的宽字符函数 API。

表 2–4 libc 中的宽字符类

库例程	说明
wctype()	定义字符类
wctrans()	定义字符映射

下表列出 libc 中的修改和查询语言环境。

表 2–5 libc 中的修改和查询语言环境

库例程	说明
setlocale()	修改和查询程序的语言环境

下表列出 libc 中的查询语言环境数据。

表 2–6 libc 中的查询语言环境数据

库例程	说明
nl_langinfo()	获取当前语言环境的语言和文化信息
localeconv()	获取当前语言环境的货币和数字格式信息

下表描述了 libc 中的字符分类函数 API。

表 2–7 libc 中的字符分类和拼写

库例程	说明
isalpha()	字符是字母吗?
isupper()	字符是大写吗?
islower()	字符是小写吗?

表 2–7 `libc` 中的字符分类和拼写 （续下）

库例程	说明
`isdigit()`	字符是数字吗?
`isxdigit()`	字符是十六进制数字吗?
`isalnum()`	字符是字母或数字吗?
`isspace()`	字符是空格吗?
`ispunct()`	字符是标点符号吗?
`isprint()`	字符是可打印字符吗?
`iscntrl()`	字符是控制字符吗?
`isascii()`	字符是 ASCII 字符吗?
`isgraph()`	字符是可见字符吗?
`isphonogram()`	宽字符是音标吗?
`isideogram()`	宽字符是表意符号吗?
`isenglish()`	宽字符在辅助代码集的英语字母表中吗?
`isnumber()`	宽字符是辅助代码集中的数字吗?
`isspecial()`	特殊宽字符在辅助代码集中吗?
`iswalpha()`	宽字符是字母吗?
`iswupper()`	宽字符是大写吗?
`iswlower()`	宽字符是小写吗?
`iswdigit()`	宽字符是数字吗?
`iswxdigit()`	宽字符是十六进制数字吗?
`iswalnum()`	宽字符是字母字符或数字吗?
`iswspace()`	宽字符是空白吗?
`iswpunct()`	宽字符是标点符号吗?
`iswprint()`	宽字符是可打印字符吗?
`iswgraph()`	宽字符是可见字符吗?
`iswcntrl()`	宽字符是控制字符吗?
`iswascii()`	宽字符是 ASCII 字符吗?
`toupper()`	将小写字符转换为大写字符。
`tolower()`	将大写字符转换为小写字符。

表 2–7 libc 中的字符分类和拼写　　(续下)

库例程	说明
towupper()	将小写宽字符转换为大写宽字符。
towlower()	将大写宽字符转换为小写宽字符。
towctrans()	宽字符映射。

下表描述了 libc 中的字符排序函数 API。

表 2–8 libc 中的字符排序

库例程	说明
strcoll()	排序字符串
strxfrm()	变换字符串以便进行比较
wcscoll()	排序宽字符串
wcsxfrm()	变换宽字符串以便进行比较

下表描述了 libc 中的货币处理函数 API。

表 2–9 libc 中的货币格式

库例程	说明
localeconv()	获取当前语言环境的货币格式信息
strfmon()	将货币值转换为字符串表示形式

下表描述了 libc 中的日期和时间格式。

表 2–10 libc 中的日期和时间格式

库例程	说明
getdate()	转换用户格式日期和时间。
strftime()	将日期和时间转换为字符串表示形式。%u 转换函数符合《X/Open CAE 规范：系统接口和头》第 4 期，第 2 版。此函数将工作日表示为十进制数字 [1,7]，其中 1 现在表示周一。
strptime()	日期和时间转换。

下表描述了 libc 中的多字节处理函数 API。

表 2–11 `libc` 中的多字节处理

库例程	说明
`btowc()`	单字节到宽字符转换
`mbrlen()`	获取字符中的字节数（可重新开始）
`mbsinit()`	确定转换对象状态
`mbrtowc()`	将字符转换为宽字符代码（可重新开始）
`mbsrtowcs()`	将字符串转换为宽字符串（可重新开始）
`mblen()`	获取字符中的字节数
`mbtowc()`	将字符转换为宽字符代码
`mbstowcs()`	将字符串转换为宽字符串

下表描述了 `libc` 中的宽字符和字符串处理。

表 2–12 `libc` 中的宽字符和字符串处理

库例程	说明
`wcsncat()`	并置宽字符串，并置长度为 *n*
`wsdup()`	复制宽字符串
`wcscmp()`	比较宽字符串
`wcsncmp()`	比较宽字符串，比较长度为 *n*
`wcscpy()`	复制宽字符串
`wcsncpy()`	复制宽字符串，复制长度为 *n*
`wcschr()`	在宽字符串中查找字符
`wcsrchr()`	从右边开始在宽字符串中查找字符
`wcslen()`	获取宽字符串的长度
`wscol()`	返回宽字符串的显示宽度
`wcsspn()`	返回一个宽字符串在另一个宽字符串中的跨度
`wcscspn()`	返回一个宽字符串不在另一个宽字符串中的跨度
`wcspbrk()`	返回指向一个位于另一个宽字符串中的宽字符串的指针
`wcstok()`	在整个宽字符串中移动标记
`wscwcs()`	在宽字符串中查找字符串
`wcstombs()`	将宽字符串转换为多字节字符串

表 2–12 libc 中的宽字符和字符串处理 *(续下)*

库例程	说明
wctomb()	将宽字符转换为多字节字符
wcwidth()	确定宽字符的列位置数
wcswidth()	确定宽字符串的列位置数
wctob()	宽字符到单字节转换
wcrtomb()	将宽字符代码转换为字符（可重新开始）
wcstol()	将宽字符串转换为长整数
wcstoul()	将宽字符串转换为无符号长整数
wcstod()	将宽字符串转换为双精度
wcsrtombs()	将宽字符串转换为字符串（可重新开始）
wcscat()	并置宽字符串

下表描述了 libc 中的格式化宽字符输入和输出。

表 2–13 libc 中的格式化宽字符输入和输出

库例程	说明
wsprintf()	根据格式生成宽字符串
wsscanf()	格式化输入转换
fwprintf()	打印格式化宽字符输出
fwscanf()	转换格式化宽字符输入
wprintf()	打印格式化宽字符输出
wscanf()	转换格式化宽字符输入
swprintf()	打印格式化宽字符输出
swscanf()	转换格式化宽字符输入
vfwprintf()	stdarg 参数列表的宽字符格式化输出
vswprintf()	stdarg 参数列表的宽字符格式化输出

下表描述 libc 中的宽字符串函数 API。

表 2–14 宽字符串libc

库例程	说明
wscasecmp()	比较宽字符串，忽略大小写差异
wsncasecmp()	进程代码串操作
wcsstr()	查找宽字符子串
wmemchr()	在内存中查找宽字符
wmemcmp()	在内存中比较宽字符
wmemcpy()	在内存中复制宽字符
wmemmove()	在具有重叠区域的内存中复制宽字符
wmemset()	在内存中设置宽字符

下表描述了 libc 中的宽字符输入和输出。

表 2–15 libc 中的宽字符输入和输出

库例程	说明
fgetwc()	从流中获取多字节字符，并转换为宽字符
getwchar()	从 stdin 中获取多字节字符，并转换为宽字符
fgetws()	从流中获取多字节串，并转换为宽字符
getws()	从 stdin 中获取多字节串，并转换为宽字符
fputwc()	将宽字符转换为多字节字符，并放入流中
fwide()	设置流定向
putwchar()	将宽字符转换为多字节字符，并放入 stdin 中
fputws()	将宽字符转换为多字节串，并放入流中
putws()	将宽字符转换为多字节串，并放入 stdin 中
ungetwc()	将宽字符推送回输入流中。

genmsg 公用程序

新的 genmsg 公用程序可以与 catgets() 系列函数一起使用，以创建国际化的源信息目录。 该公用程序在源程序文件中检查对 catgets 中函数的调用，然后根据找到的信息生成源信息目录。 例如：

```
% cat example.c
    ...
    /* NOTE: %s is a file name */
    printf(catgets(catd, 5, 1, "%s cannot be opened."));
    /* NOTE: "Read" is a past participle, not a present

            tense verb */
    printf(catgets(catd, 5, 1, "Read"));
    ...
% genmsg -c NOTE example.c
已经创建下列文件。
            new msg file = "example.c.msg"
% cat example.c.msg
$quote "
$set 5
1             "%s cannot be opened"
    /* NOTE: %s is a file name */
2             "Read"
    /* NOTE: "Read" is a past participle, not a present
            tense verb */
```

在上面的示例中，对源文件 example.c 运行 genmsg，产生一个名为 example.c.msg 的源信息目录。 带 NOTE 参数的 -c 选项使 genmsg 在目录中包含注释。 如果源程序中的注释包含指定的字符串，则该注释在信息目录中出现在下一个从对 catgets 的调用中提取的字符串的后面。

可以使用 genmsg 对信息集中的信息进行自动编号。

有关详细信息，请参见 genmsg(1) 手册页。

要生成格式化信息目录文件，请使用 gencat (1) 公用程序。

有关可移植信息文件（.po 文件）的信息提取公用程序以及如何从 .po 文件生成信息对象文件（.mo 文件）的信息，请分别参见 xgettext(1) 和 msgfmt(1) 手册页。

用户定义的和用户可扩展的代码转换

Solaris 用户可以使用 geniconvtbl 公用程序来创建用户定义的代码集转换器。

该公用程序使得能够通过标准系统公用程序和界面（如 iconv(1) 和 iconv(3C)）进行用户定义的和用户可定制的代码集转换。该特征增强了应用程序处理不兼容的数据类型特别是从专用或传统应用程序生成的数据的能力。 还支持对现有的 Solaris 代码集转换的修改。

在 geniconvtbl (1) 和 geniconvtbl(4) 手册页中可以找到更多的详细信息和示例。还可以在 /usr/lib/iconv/geniconvtbl/srcs/ 目录中得到该公用程序的输入源文件样例，以便参考。

在如 geniconvtbl(1) 手册页中所指定的那样准备并放置用户定义的代码转换之后，用户便可以从 32 位和 64 位 Solaris 操作环境的 iconv(1) 公用程序和 iconv(3C) 函数使用这些代码转换了。

第 3 章

Solaris 9 环境中的本地化

本部分讨论 Solaris 9 环境中包含的几种本地化特征。

- 第 53 页 “本地化的软件支持”
- 第 55 页 “支持的语言环境”
- 第 61 页 “语言环境的多键书写序列”
- 第 62 页 “Solaris 9 产品中的键盘支持”

本地化的软件支持

本部分包含有关 Solaris 9 语言环境软件包、Solaris 9 CD-ROM 上的本地化内容、界面中的本地化功能以及语言启用的信息。

Solaris 9 语言环境软件包摘要

所有 Solaris 9 语言环境软件包分为两类。

第一类用于不完全语言环境，它是语言环境的启用程序。在系统上安装不完全语言环境后，用户可以在目标语言环境中输入、显示、打印文本以及运行应用程序，而 Solaris 中的 OS/GUI 信息均为英语，所有不完全语言环境软件包都可以从 Solaris Software CD 中得到。日语和亚洲语言不完全语言环境将根据语言进行封装，其它不完全语言环境则根据地区进行封装。

第二类用于完全语言环境软件包。这些软件包包括软件信息的翻译、联机帮助文件、可选字体和语言的特定特征。完全语言环境软件包为许多语言提供了全套语言特征。所有基于下列语言的语言环境均是完全语言环境：

- 德语
- 法语

- 西班牙语
- 瑞典语
- 意大利语
- 日语
- 朝鲜语
- 简体中文
- 繁体中文

完全语言环境软件包根据语言进行封装，并可以在 Language CD 中得到。

注意：为了使完全语言环境起作用，必须安装不完全语言环境软件包（语言环境启用程序）。

在 Solaris 安装过程中，会提示您选择需要支持的地区。在安装完成后可用的语言环境支持取决于在该阶段所做的选择。不完全语言环境从具有 Solaris 9 操作环境的 Solaris Software CD-ROM 安装，而完全语言环境则从 Language CD 安装。如果不需要完全语言环境支持，则可以在安装过程中跳过从 Language CD-ROM 进行的安装。请注意，所安装的缺省语言环境为英语语言环境。

本版本中的附加语言环境

本版本的新的不完全语言环境包括为俄语和波兰语添加的 UTF–8 语言环境、两个新的加泰罗尼亚文语语言环境、一个新的泰语语言环境、一个新的印度语语言环境、两个新的繁体中文语言环境和一个新的简体中文语言环境。这些语言环境的名称是：

- `ar_EG.UTF-8`
- `ca_ES.ISO8859-1`
- `ca_ES.ISO8859-15`
- `fi_FI.UTF-8`
- `fr_BE.UTF-8`
- `pl_PL.UTF-8`
- `pt_BR.UTF-8`
- `ru_RU.UTF-8`
- `tr_TR.UTF-8`
- `th_TH.UTF-8`
- `hi_IN.UTF-8`
- `zh_HK.BIG5HK`

  这是一个繁体中文（香港特别行政区）Big5–HKSCS 语言环境。如果是从 Language CD 安装繁体中文信息软件包，则它是完全语言环境。
- `zh_HK.UTF-8`

这是一个繁体中文（香港特别行政区）UTF—8 语言环境。如果繁体中文信息软件包从 Language CD 安装，则它将是完全语言环境。

- zh_CN.GB18030

  这是一个简体中文 GB18030 语言环境。 如果是从 Language CD 安装简体中文信息软件包，则它是完全语言环境。

支持的语言环境

下列各表列出了 Solaris 9 环境所支持的所有语言环境。语言环境名称已被更新，以便与国际命名标准保持一致。

表 3–1 亚洲

语言环境	用户界面	地区	代码集	语言支持
hi_IN.UTF-8	英语	印度	UTF-8[1]	印地语 (UTF-8) Unicode 3.1
ja	日语	日本	eucJP[2]	日语 (EUC)
				JIS X 0201-1976
				JIS X 0208-1990
				JIS X 0212-1990
ja_JP.eucJP	日语	日本	eucJP	日语 (EUC)
				JIS X 0201-1976
				JIS X 0208-1990
				JIS X 0212-1990
ja_JP.PCK	日语	日本	PCK[3]	日语 (PC kanji)
				JIS X 0201-1976
				JIS X 0208-1990
ja_JP.UTF-8	日语	日本	UTF-8	日语 (UTF-8) Unicode 3.1
ko_KR.EUC	朝鲜语	韩国	1001	朝鲜语 (EUC) KS X 1001
ko_KR.UTF-8	朝鲜语	韩国	UTF-8	朝鲜语 (UTF-8) Unicode 3.1

UTF-8 是在 ISO/IEC 10646–1:2000 中定义的 UTF-8，同时也是在 Unicode 3.1 定义的 UTF-8。

eucJP 表示日语 EUC 代码集。 ja_JP.eucJP 语言环境规范符合 UI_OSF 日语环境执行协议 1.1 版，ja 语言环境符合过去的 Solaris 版本的传统规范。

PCK 也称作 Shift_JIS (SJIS)。

表 3–1 亚洲 （续下）

语言环境	用户界面	地区	代码集	语言支持
th_TH.UTF-8	英文	泰国	UTF-8	泰语 (UTF-8) Unicode 3.1
th_TH.TIS620	英文	泰国	TIS620.2533	泰语 TIS620.2533
zh_CN.EUC	简体中文	中国大陆地区	gb2312[4]	简体中文 (EUC) GB2312-1980
zh_CN.GBK	简体中文	中国	GBK [5]	简体中文 (GBK)
zh_CN.GB18030	简体中文	中国大陆地区	GB18030–2000	简体中文 (GB18030–2000) GB18030–2000
zh_CN.UTF-8	简体中文	中国大陆地区	UTF-8	简体中文 (UTF-8) Unicode 3.1
zh_HK.BIG5HK	繁体中文	香港特别行政区	Big5+HKSCS	繁体中文 (BIG5+HKSCS)
zh_HK.UTF-8	繁体中文	香港特别行政区	UTF-8	繁体中文 (UTF-8) Unicode 3.1
zh_TW.EUC	繁体中文	中国台湾地区	cns11643	繁体中文 (EUC) CNS 11643-1992
zh_TW.BIG5	繁体中文	中国台湾地区	BIG5	繁体中文 (BIG5)
zh_TW.UTF-8	繁体中文	中国台湾地区	UTF-8	繁体中文 (UTF-8) Unicode 3.1

gb2312 表示简体中文 EUC 代码集，其中包含 GB 1988–80 和 GB 2312–80。

GBK 表示 GB 扩展。 它包括所有的 GB 2312–80 字符和 ISO/IEC 10646–1 中的所有统一汉字字符以及日语平假名和片假名字符。它还包括许多中文、日语和朝鲜语字符集中的字符以及 ISO/IEC 10646–1 中的许多字符。

表 3–2 澳大拉西亚

语言环境	用户界面	地区	代码集	语言支持
en_AU.ISO8859-1	英语	澳大利亚	ISO8859-1	英语（澳大利亚）
en_NZ.ISO8859-1	英语	新西兰	ISO8859-1	英语（新西兰）

表 3–3 中美洲

语言环境	用户界面	地区	代码集	语言支持
es_CR.ISO8859-1	西班牙语	哥斯达黎加	ISO8859-1	西班牙语（哥斯达黎加）
es_GT.ISO8859-1	西班牙语	危地马拉	ISO8859-1	西班牙语（危地马拉）
es_NI.ISO8859-1	西班牙语	尼加拉瓜	ISO8859-1	西班牙语（尼加拉瓜）
es_PA.ISO8859-1	西班牙语	巴拿马	ISO8859-1	西班牙语（巴拿马）
es_SV.ISO8859-1	西班牙语	萨尔瓦多	ISO8859-1	西班牙语（萨尔瓦多）

表 3–4 中欧

语言环境	用户界面	地区	代码集	语言支持
cs_CZ.ISO8859-2	英语	捷克共和国	ISO8859-2	捷克语（捷克共和国）
de_AT.ISO8859-1	德语	奥地利	ISO8859-1	德语（奥地利）
de_AT.ISO8859-15	德语	奥地利	ISO8859-15	德语（奥地利，ISO8859-15 - Euro）
de_CH.ISO8859-1	德语	瑞士	ISO8859-1	德语（瑞士）
de_DE.UTF-8	德语	德国	UTF-8	德语（德国，Unicode 3.1）
de_DE.ISO8859-1	德语	德国	ISO8859-1	德语（德国）
de_DE.ISO8859-15	德语	德国	ISO8859-15	德语（德国，ISO8859-15 - Euro）
fr_CH.ISO8859-1	法语	瑞士	ISO8859-1	法语（瑞士）
hu_HU.ISO8859-2	英语	匈牙利	ISO8859-2	匈牙利语（匈牙利）
pl_PL.ISO8859-2	英语	波兰	ISO8859-2	波兰语（波兰）
pl_PL.UTF-8	英语	波兰	UTF-8	波兰语（波兰，Unicode 3.1）
sk_SK.ISO8859-2	英语	斯洛伐克	ISO8859-2	斯洛伐克语（斯洛伐克）

表 3–5 东欧

语言环境	用户界面	地区	代码集	语言支持
bg_BG.ISO8859-5	英语	保加利亚	ISO8859-5	保加利亚语（保加利亚）
et_EE.ISO8859-15	英语	爱沙尼亚	ISO8859-15	爱沙尼亚语（爱沙尼亚）
hr_HR.ISO8859-2	英语	克罗地亚	ISO8859-2	克罗地亚语（克罗地亚）
lt_LT.ISO8859-13	英语	立陶宛	ISO8859-13	立陶宛语（立陶宛）
lv_LV.ISO8859-13	英语	拉脱维亚	ISO8859-13	拉脱维亚语（拉脱维亚）
mk_MK.ISO8859-5	英语	马其顿	ISO8859-5	马其顿语（马其顿）
ro_RO.ISO8859-2	英语	罗马尼亚	ISO8859-2	罗马尼亚语（罗马尼亚）

表 3–5 东欧　　(续下)

语言环境	用户界面	地区	代码集	语言支持
ru_RU.KOI8-R	英语	俄罗斯	KOI8-R	俄语（俄罗斯，KOI8-R）
ru_RU.ANSI1251	英语	俄罗斯	ansi-1251	俄语（俄罗斯，ANSI 1251）
ru_RU.ISO8859-5	英语	俄罗斯	ISO8859-5	俄语（俄罗斯）
ru_RU.UTF-8	英语	俄罗斯	UTF-8	俄语（俄罗斯，Unicode 3.1）
sh_BA.ISO8859-2@bosnia	英语	波斯尼亚	ISO8859-2	波斯尼亚语（波斯尼亚）
sl_SI.ISO8859-2	英语	斯洛文尼亚	ISO8859-2	斯洛文尼亚语（斯洛文尼亚）
sq_AL.ISO8859-2	英语	阿尔巴尼亚	ISO8859-2	阿尔巴尼亚语（阿尔巴尼亚）
sr_YU.ISO8859-5	英语	塞尔维亚	ISO8859-5	塞尔维亚语（塞尔维亚）
tr_TR.ISO8859-9	英语	土耳其	ISO8859-9	土耳其语（土耳其）
tr_TR.UTF-8	英语	土耳其	UTF-8	土耳其语（土耳其，Unicode 3.1）

表 3–6 中东

语言环境	用户界面	地区	代码集	语言支持
He	英语	以色列	ISO8859-8	希伯来语（以色列）

表 3–7 北非

语言环境	用户界面	地区	代码集	语言支持
ar_EG.UTF-8	英语	埃及	UTF-8	阿拉伯语（埃及）
Ar	英语	埃及	ISO8859-6	阿拉伯语（埃及）

表 3–8 北美洲

语言环境	用户界面	地区	代码集	语言支持
en_CA.ISO8859-1	英语	加拿大	ISO8859-1	英语（加拿大）

表 3–8 北美洲 *（续下）*

语言环境	用户界面	地区	代码集	语言支持
en_US.ISO8859-1	英语	美国	ISO8859-1	英语（美国）
en_US.ISO8859-15	英语	美国	ISO8859-15	英语（美国，ISO8859-15 - Euro）
en_US.UTF-8	英语	美国	UTF-8	英语（美国，Unicode 3.1）
fr_CA.ISO8859-1	法语	加拿大	ISO8859-1	法语（加拿大）
es_MX.ISO8859-1	西班牙语	墨西哥	ISO8859–1	西班牙语（墨西哥）

表 3–9 北欧

语言环境	用户界面	地区	代码集	语言支持
da_DK.ISO8859-1	英语	丹麦	ISO8859–1	丹麦语（丹麦）
da_DK.ISO8859-15	英语	丹麦	ISO8859–15	丹麦语（丹麦，ISO8859–15–Euro）
fi_FI.ISO8859-1	英语	芬兰	ISO8859–1	芬兰语，Unicode 3.1）
fi_FI.ISO8859-15	英语	芬兰	ISO8859–15	芬兰语（芬兰，ISO8859–15–Euro）
fi_FI.UTF-8	英语	芬兰	UTF-8	芬兰语（芬兰）
is_IS.ISO8859-1	英语	冰岛	ISO8859–1	冰岛语（冰岛）
no_NO.ISO8859-1@bokmal	英语	挪威	ISO8859–1	挪威语（挪威-博克马尔语）
no_NO.ISO8859-1@nyorsk	英语	挪威	ISO8859–1	挪威语（挪威-尼诺斯克语）
sv_SE.ISO8859-1	瑞典语	瑞典	ISO8859–1	瑞典语（瑞典）
sv_SE.ISO8859-15	瑞典语	瑞典	ISO8859–15	瑞典语（瑞典，ISO8859–15–Euro）
sv_SE.UTF-8	瑞典语	瑞典	UTF-8	瑞典语（瑞典，Unicode 3.1）

表 3–10 南美洲

语言环境	用户界面	地区	代码集	语言支持
es_AR.ISO8859-1	西班牙语	阿根廷	ISO8859-1	西班牙语（阿根廷）
es_BO.ISO8859-1	西班牙语	玻利维亚	ISO8859-1	西班牙语（玻利维亚）
es_CL.ISO8859-1	西班牙语	智利	ISO8859-1	西班牙语（智利）
es_CO.ISO8859-1	西班牙语	哥伦比亚	ISO8859-1	西班牙语（哥伦比亚）

表 3–10 南美洲 （续下）

语言环境	用户界面	地区	代码集	语言支持
es_EC.ISO8859-1	西班牙语	厄瓜多尔	ISO8859-1	西班牙语（厄瓜多尔）
es_PE.ISO8859-1	西班牙语	秘鲁	ISO8859-1	西班牙语（秘鲁）
es_PY.ISO8859-1	西班牙语	巴拉圭	ISO8859-1	西班牙语（巴拉圭）
es_UY.ISO8859-1	西班牙语	乌拉圭	ISO8859-1	西班牙语（乌拉圭）
es_VE.ISO8859-1	西班牙语	委内瑞拉	ISO8859-1	西班牙语（委内瑞拉）
pt_BR.ISO8859-1	英语	巴西	ISO8859-1	葡萄牙语（巴西）
pt_BR.UTF-8	英语	巴西	UTF-8	葡萄牙语（巴西，Unicode 3.1）

表 3–11 南欧

语言环境	用户界面	地区	代码集	语言支持
ca_ES.ISO8859-1	英语	西班牙	ISO8859-1	加泰罗尼亚语（西班牙）
ca_ES.ISO8859-15	英语	西班牙	ISO8859-15	加泰罗尼亚语（西班牙，ISO8859-15 - Euro）
el_GR.ISO8859-7	英语	希腊	ISO8859-7	希腊语（希腊）
es_ES.ISO8859-1	西班牙语	西班牙	ISO8859-1	西班牙语（西班牙）
es_ES.ISO8859-15	西班牙语	西班牙	ISO8859-15	西班牙语（西班牙，ISO8859-15 - Euro）
es_ES.UTF-8	西班牙语	西班牙	UTF-8	西班牙语（西班牙，Unicode 3.1）
it_IT.ISO8859-1	意大利语	意大利	ISO8859-1	意大利语（意大利）
it_IT.ISO8859-15	意大利语	意大利	ISO8859-15	意大利语（意大利，ISO8859-15 - Euro）
it_IT.UTF-8	意大利语	意大利	UTF-8	意大利语（意大利，Unicode 3.1）
pt_PT.ISO8859-1	英语	葡萄牙	ISO8859-1	葡萄牙语（葡萄牙）
pt_PT.ISO8859-15	英语	葡萄牙	ISO8859-15	葡萄牙语（葡萄牙，ISO8859-15 - Euro）

表 3–12 西欧

语言环境	用户界面	地区	代码集	语言支持
en_GB.ISO8859-1	英语	大不列颠	ISO8859-1	英语（大不列颠）

表 3–12 西欧 *（续下）*

语言环境	用户界面	地区	代码集	语言支持
`en_IE.ISO8859-1`	英语	爱尔兰	ISO8859-1	英语（爱尔兰）
`fr_BE.ISO8859-1`	法语	比利时-瓦龙语	ISO8859-1	法语（比利时-瓦龙语，Unicode 3.1）
`fr_BE.UTF-8`	法语	比利时-瓦龙语	UTF-8	法语（比利时-瓦龙语，Unicode 3.1）
`fr_FR.ISO8859-1`	法语	法国	ISO8859-1	法语（法国）
`fr_FR.UTF-8`	法语	法国	UTF-8	法语（法国，Unicode 3.1）
`nl_BE.ISO8859-1`	英语	比利时-佛兰芒语	ISO8859-1	荷兰语（比利时-佛兰芒语）
`nl_NL.ISO8859-1`	英语	荷兰	ISO8859-1	荷兰语（荷兰）

语言环境的多键书写序列

许多 Solaris 语言环境，特别是欧洲和 Unicode 语言环境允许使用所谓的“死键序列”（也称为书写键序列）输入各种字符。

书写键序列输入用于输入带有发音符的字符以及其它在键盘上未显示的字符。

下表显示了几个书写键序列的示例。有关书写键序列的更完整信息，请参见第 96 页 “英语/欧洲语言输入模式” 。

表 3–13 使用书写键创建的发音字符

标记	书写键组合	示例
分音符	″	书写 A “ —> 带分音符的 A
抑扬符	V	书写 Z v —> 带抑扬符的 Z
短音符	u	书写 G u —> 带短音符的 G
Ogonek	a	书写 A a —> 带 Ogonek 的 A
下加符	,	书写 K , —> 带下加符的K
注册标记	R O	书写 R O —> 注册标记
倒置叹号	! !	书写 ! ! —> 倒置惊叹号

注意：如果当前语言环境的代码集没有相应的字符，则不能使用书写序列。例如，因为在 ISO8859–1 中没有带caron的 `z`，所以在 `en_US.ISO8859-1` 语言环境中不可能输入带caron的 `z`。

Solaris 9 产品中的键盘支持

Solaris 识别并支持各种键盘，这些键盘具有为特定地区设置的不同键盘布局，以及支持适用于 SunSPARC 和 Intel 体系结构(IA) 平台的布局。Solaris 9 支持下表中列出的地区键盘：

表 3–14 对地区键盘的支持

地区	国家/地区	Sun 键盘（4/5/5c 型）	Sun 键盘（6 型）	PC 键盘
亚洲	日本	X	X	X
	韩国	X	X	X
	中国台湾地区	X	X	X
欧洲	比利时	X	X	X
	捷克共和国	X		X
	丹麦	X	X	X
	芬兰		X	
	法国	X	X	X
	德国	X	X	X
	大不列颠	X	X	X
	希腊	X		X
	匈牙利	X		X
	意大利	X	X	X
	拉脱维亚	X		X
	立陶宛	X		X
	荷兰	X	X	X
	挪威	X	X	X

表 3–14 对地区键盘的支持 (续下)

地区	国家/地区	Sun 键盘（4/5/5c 型）	Sun 键盘（6 型）	PC 键盘
	波兰	X		X
	葡萄牙	X	X	X
	俄罗斯	X	X	X
	西班牙	X	X	X
	瑞典	X	X	X
	瑞士（法语）	X	X	X
	瑞士（德语）	X	X	X
	土耳其	X	X	X
美洲	加拿大（法语）	X	X	X
	拉丁美洲（西班牙语）	X		
	美国	X	X	X
中东	阿拉伯语	X	X	

对于键盘布局符合国际标准的地区（如中国），则使用为美国提供的键盘布局支持输入该语言环境的字符。基础键盘映射是相同的。某些国家（如日本、土耳其和瑞士）具有多种键盘，原因是使用多种语言或者存在多种键盘布局。

Sun 4、5 和 5c 型键盘通过 Mini DIN 8 针连接使用 Sun I/O 接口。Sun 6 型键盘具有两个接口版本：

- 通过 Mini DIN 8 针连接 Sun I/O 接口
- USB

Sun 键盘类型印在每个 Sun 键盘的背面。

PC 键盘使用各种不同的接口，如 PS/2 或 USB。

在 SPARC 系统上更改键盘

用户可以使用大多数 Sun 4、5 和 5c 型键盘底部的 DIP 开关设置来更改 Solaris 产品中的键盘布局。可用于 DIP 开关设置的键盘类型、名称和相应的布局 ID 的列表位于 `/usr/openwin/share/etc/keytables/keytable.map` 文件中。

注意：用户不能更改 6 型键盘的布局，因为在键盘的背面没有 DIP 开关。某些 5 型和 5c 型键盘（如美国键盘、美国/UNIX 键盘和日语键盘）具有跳线，而没有 DIP 开关。对于 SPARC 和 IA 这两种平台来说，在 Solaris 9 操作环境中没有捆绑的公用程序或工具（除标准 UNIX 工具如 `xmodmap`(1) 以外）用来切换键盘。

下面是 4、5 和 5c 型键盘的布局 ID 值列表。（1 = 向上切换，0 = 向下切换）。

表 3–15 4、5 和 5c 型键盘的布局

DIP 开关	键盘（Keytable 文件）	二进制设置
0	美国 (`US4.kt`)	000000
1	美国 (`US4.kt`)	000001
2	比利时 (`FranceBelg4.kt`)	000010
3	加拿大 (`Canada4.kt`)	000011
4	丹麦 (`Denmark4.kt`)	000100
5	德国 (`Germany4.kt`)	000101
6	意大利 (`Italy4.kt`)	000110
7	荷兰 (`Netherland4.kt`)	000111
8	挪威 (`Norway4.kt`)	001000
9	葡萄牙（`Portugal4.kt`)	001001
10 (0x0a)	拉丁美洲/西班牙语 (`SpainLatAm4.kt`)	001010
11 (ox0b)	瑞典 (`SwedenFin4.kt`)	001011
12 (0x0c)	瑞士/法语 (`Switzer_Fr4.kt`)	001100
13 (0x0d)	瑞士/德语 (`Switzer_Ge4.kt`)	001101
14 (0x0e)	大不列颠 (`UK4.kt`)	001110
16 (0x10)	韩国 (`Korea4.kt`)	010000
17 (0x11)	中国台湾地区 (`Taiwan4.kt`)	010001
33 (0x21)	美国 (`US5.kt`)	100001
34 (0x22)	美国/UNIX (`US_UNIX5.kt`)	100010
35 (0x23)	法国 (`France5.kt`)	100011
36 (0x24)	丹麦 (`Denmark5.kt`)	100100
37 (0x25)	德国 (`Germany5.kt`)	100101

表 3–15 4、5 和 5c 型键盘的布局　　(续下)

DIP 开关	键盘（Keytable 文件）	二进制设置
38 (0x26)	意大利 (Italy5.kt)	100110
39 (0x27)	荷兰 (Netherland5.kt)	100111
40 (0x28)	挪威 (Norway5.kt)	101000
41 (0x29)	葡萄牙 (Portugal5.kt)	101001
42 (0x2a)	西班牙 (Spain5.kt)	101010
43 (0x2b)	瑞典 (Sweden5.kt)	101011
44 (0x2c)	瑞士/法语 (Switzer_Fr5.kt)	101101
45 (0x2d)	瑞士/德语 (Switzer_Ge5.kt)	101110
46 (0x2e)	大不列颠 (UK5.kt)	101111
47 (0x2f)	韩国 (Korea5.kt)	101111
48 (0x30)	中国台湾地区 (Taiwan5.kt)	110000
49 (0x31)	日本 (Japan5.kt)	110001
50 (0x32)，参见 63 (0x3f)	加拿大/法语 (Canada_Fr5.kt)	110010
51 0(x33)	匈牙利 (Hungary5.kt)	110011
52 (0x34	波兰 (Poland5.kt)	110100
53 (0x35)	捷克语 (Czech5.kt)	110101
54 (0x36)	俄罗斯 (Russia5.kt)	110110
55 (0x37)	拉脱维亚 (Latvia5.kt)	110111
56 (0x38)，参见 62 (0x3e)	土耳其-Q5 (TurkeyQ5.kt)	111000
57 (0x39)	希腊 (Greece5.kt)	111001
58 (0x3a)	阿拉伯语 (Arabic5.kt)	111011
59 (0x3b)	立陶宛 (Lithuania5.kt)	111010
60 (0x3c)	比利时 (Belgian5.kt)	111100
62 (0x3e)	土耳其-F5 (TurkeyF5.kt)	111110
63 (0x3f)	加拿大/法语 (Canada_Fr5_TBITS5.kt)	111111

带有 4 的 Keytable 文件名用于 4 型键盘。带有 5 的 Keytable 文件名用于 5 型键盘。

将布局从一种键盘布局更改为另一种布局（如捷克语）需要以下步骤：

1. 从该表或从 /usr/openwin/share/etc/keytables/keytable.mp 文件中查找正确的 DIP 开关 ID（或布局 ID）。keytable.mp 文件中的布局 ID 值是一个十进制值。

   对于捷克语，布局 ID 为十进制的 53（即十六进制的 0x35）。
2. 将布局 ID 转换为二进制，或者使用上表表列中正确的"二进制设置"值。对于基转换，可能会用到计算器公用程序，如 dtcalc(1)。

   捷克语键盘的正确二进制值是 110101。
3. 成为超级用户。关机并切断系统电源。
4. 使用第 2 步中的二进制值来更改键盘背面的 DIP 开关设置。

   第一个 DIP 开关位于您的左侧。对于"1"，将开关向上切换；对于"0"，将开关向下切换。

   捷克语键盘的二进制值 110101 对应于： 上 上 下 上 下 上。
5. 打开电源并引导系统以备使用。

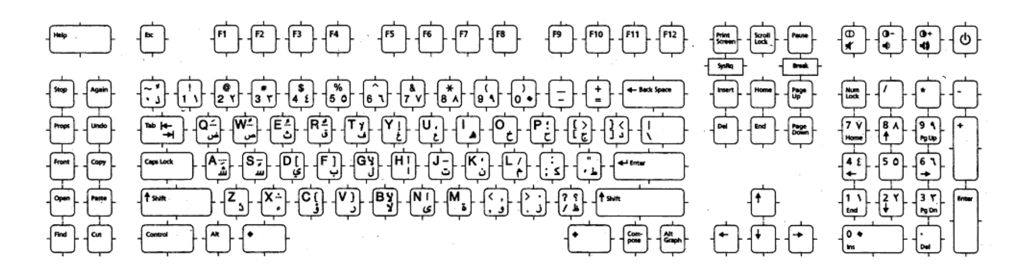
注意：与 4 型键盘不同，5 型和 5c 型键盘只有五个 DIP 开关。对于 5 型和 5c 型键盘，请忽略第一个二进制数字。例如，对于捷克语 5c 型键盘，正确的 DIP 开关设置仅使用最后五位数字 10101，即为"上 下 上 下 上"。

在 Intel 系统上更改键盘

在 Intel 体系结构系统上，键盘在安装过程的 kdmconfig(1M) 阶段选定。要在安装后的任何时间更改该设置，请首先退出 GUI 桌面环境并转到命令行模式下。作为超级用户，请键入 kdmconfig 来运行程序。遵循说明来获取所需的键盘布局。

键盘布局图解

下图显示了阿拉伯语键盘。

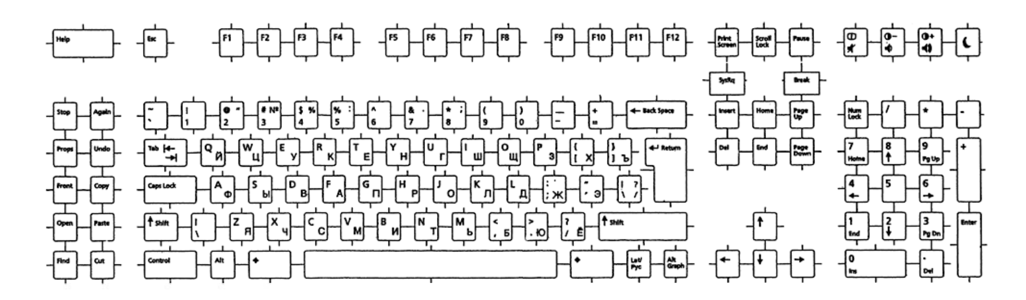


图形 3–1 阿拉伯语键盘

下图显示了比利时语键盘。

图形 3–2 比利时语键盘

下图显示了西里尔语键盘。



图形 3–3 西里尔语（俄语）键盘

下图显示了丹麦语键盘。

图形 3–4 丹麦语键盘

下图显示了芬兰语键盘。

图形 3–5 芬兰语键盘

下图显示了法语键盘。

图形 3–6 法语键盘

下图显示了德语键盘。

图形 3–7 德语键盘

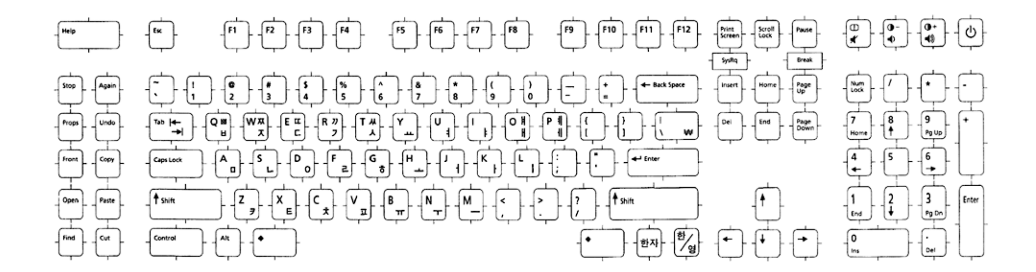
下图显示了意大利语键盘。

图形 3–8 意大利语键盘

下图显示了日语键盘。

图形 3–9 日语键盘

下图显示了朝鲜语键盘。



图形 3–10 朝鲜语键盘

下图显示了荷兰（荷兰语）键盘。

图形 3–11 荷兰（荷兰语）键盘

下图显示了挪威语键盘。

图形 3–12 挪威语键盘

下图显示了葡萄牙语键盘。

图形 3–13 葡萄牙语键盘

下图显示了西班牙语键盘。

图形 3–14 西班牙语键盘

下图显示了瑞典语键盘。

图形 3–15 瑞典语键盘

下图显示了瑞士（法语）键盘。

图形 3–16 瑞士（法语）键盘

下图显示了瑞士（德语）键盘。

图形 3–17 瑞士（德语）键盘

下图显示了繁体中文键盘。

图形 3–18 繁体中文键盘

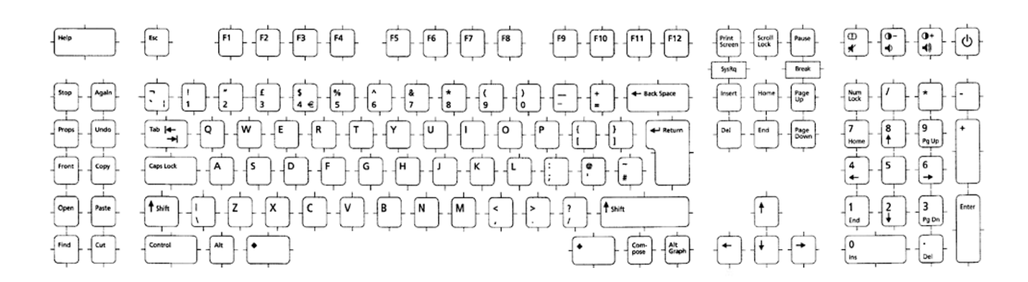
下图显示了土耳其语 F 型键盘。

图形 3–19 土耳其语 F 型键盘

下图显示了土耳其语 Q 型键盘。

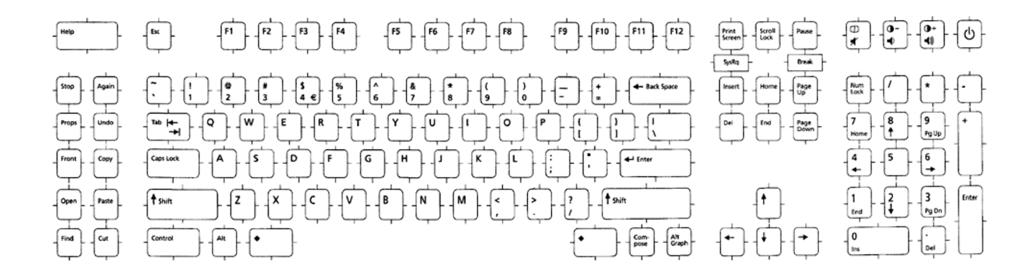
图形 3–20 土耳其语 Q 型键盘

下图显示了英国键盘。



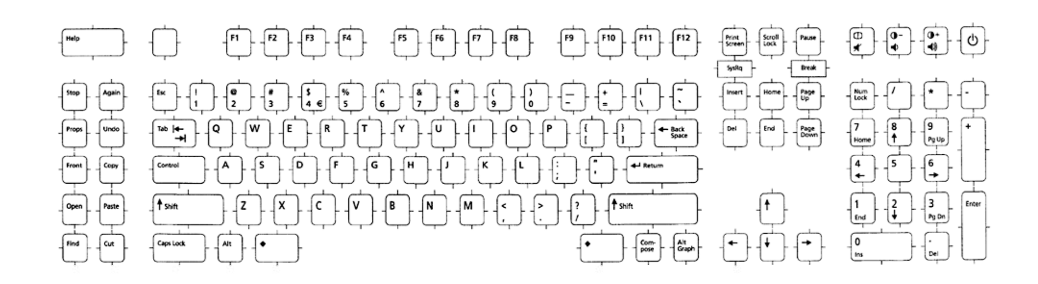
图形 3–21 英国键盘

下图显示了美国键盘。



图形 3–22 美国键盘

下图显示了美国/UNIX 键盘。



图形 3–23 美国/UNIX 键盘

第 4 章

支持的亚洲语言环境

下列部分描述亚洲受支持的语言环境：

- 第 77 页 “泰语本地化”
- 第 78 页 “简体中文本地化”
- 第 81 页 “繁体中文本地化”
- 第 84 页 “日语本地化”
- 第 87 页 “朝鲜语本地化”

亚洲受支持的语言环境

下表提供了亚洲受支持的语言环境的摘要。

表 4–1 亚洲语言环境摘要

语言	语言环境名称	描述	支持的字符集
朝鲜语	ko	朝鲜语 (EUC)	KS X 1001
	ko.UTF-8	朝鲜语 (UTF-8)	KS X 1005–1
简体中文	zh_CN.EUC	简体中文 (EUC)	GB 2312-1980
	zh_CN.GBK	简体中文 (GBK)	GBK
	zh_CN.GB18030	简体中文 (GB18030–2000)	GB18030–2000
	zh_CN.UTF-8	简体中文 (UTF-8)	Unicode 3.1

表 4–1 亚洲语言环境摘要 （续下）

语言	语言环境名称	描述	支持的字符集
繁体中文	zh_TW.EUC	繁体中文 (EUC)	CNS 11643 –1992
	zh_TW.UTF-8	繁体中文 (UTF-8)	Unicode 3.1
	zh_TW.BIG5	繁体中文 (BIG5)	BIG5
	zh_HK.BIG5HK	繁体中文 (BIG5+HKSCS)	BIG5+HKSCS
	zh_HK.UTF–8	繁体中文 (UTF-8)	Unicode 3.1
日语	ja	日语 (EUC)	JIS[1]
	ja_JP.eucJP	日语 (EUC)	JIS[2]
	ja_JP.PCK	日语 (PCK)	Unicode3.1
	ja_JP.UTF-8	日语 (UTF-8)	
泰语	th_TH.TIS620	泰文 (TIS620.2533)	TIS620.2533
	th_TH.UTF-8	泰文 (UTF-8)	Unicode 3.1
	th_TH.ISO8859-11	Thai(ISO8859-11)	ISO8859-11
印地语	hi_IN.UTF-8	印地语 (UTF-8)	Unicode 3.1

JIS X 0201-1976、JIS X 0208-1990 和 JIS X 0212-1990
JIS X 0201–1976 和 JIS X 0208–1990

简体和繁体中文的输入法辅助窗口支持

该窗口为所有中文用户提供了一个友好并且可扩展的输入法管理工具。新的输入法辅助窗口支持下列新功能和公用程序：

- 输入法切换
- 中文全角/半角字符模式切换
- 中文/英语标点模式切换
- 输入法特性设置
- 输入法选项
- GB2312/GBK/GB18030/CNS11643/Big5/HKSCS/Unicode 字符集的查找表
- 虚拟键盘

有关更详细的信息，请参见《**简体中文用户指南**》和《**繁体中文用户指南**》。

输入法辅助窗口支持所有的 UTF-8 语言环境和下列中文语言环境：

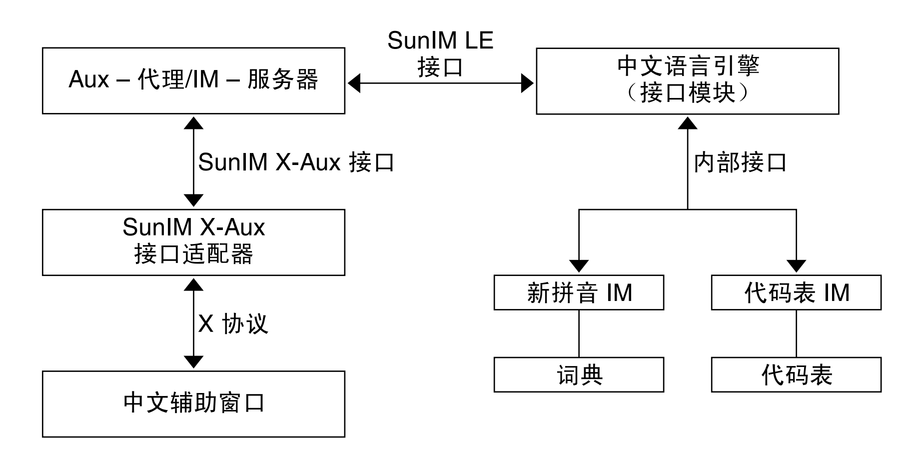
- zh/zh_CN.EUC
- zh.GBK/zh_CN.GBK
- zh.UTF-8/zh_CN.UTF-8
- zh_TW/zh_TW.EUC

- zh_TW.BIG5
- zh_TW.UTF-8
- zh_HK.BIG5HK
- zh_HK.UTF-8
- zh_CN.GB18030

支持两种输入法：

- 基于代码表的输入法（如仓颉）
- 由供应商开发的输入法（如新拼音或内码）

辅助窗口支持的接口模型如下图所示。



图形 4–1 辅助窗口支持的接口模型

泰语本地化

根据泰语信息技术规范，泰语字符序列检查方法有三种输入级别：

1. 传递级别，不进行任何输入检查。
2. 基本的输入检查级别。
3. 严格的输入检查级别。

在 Solaris 9 发行版中，缺省的输入检查级别仍为传递级别。这意味着不进行任何序列检查，这与以前的 Solaris 版本中的级别相同。 可以使用 F2 功能键在这三个级别之间进行切换：

```
传递 -> 基本 -> 严格 -> 传递
```

泰语输入法辅助窗口

泰语输入法辅助窗口支持下列新功能和公用程序：

- 在三个输入级别（传递/基本/严格）之间进行切换
- 泰语虚拟键盘

单击辅助条上的输入级别按钮以选择特定的泰语输入级别和输入检查级别。单击键盘按钮以显示泰语虚拟键盘。使用泰语虚拟键盘输入泰语字符。

简体中文本地化

Solaris 9 环境中的简体中文提供了四种语言环境：zh、zh.GBK、zh_CN.GB18030 和 zh.UTF-8。在 zh 语言环境中，EUC 方案用于对 GB2312–80 进行编码。zh.GBK 语言环境支持 GBK 代码集，它是 GB2312–80 的超集。

现在 zh_CN.GB18030 语言环境支持新的 GB18030–2000 代码集。

简体中文主要在中华人民共和国 (PRC) 和新加坡使用。

下列输入法在 zh 语言环境中受到支持：

- 新全拼
- 新双拼
- 全拼
- 双拼
- GB2312 内码
- 英语-中文
- 可选的代码表输入法
- 简体中文的输入法辅助窗口支持

下列输入法在 zh_CN.GB18030 语言环境中受到支持：

- 新全拼
- 新双拼
- 全拼
- 双拼
- GB18030–2000 内码
- 英语-中文
- 可选的代码表输入法
- 简体中文的输入法辅助窗口支持

下列输入法在 zh.GBK 和 zh.UTF-8 语言环境中同时受到支持：

- 新全拼
- 新双拼
- 全拼

- 双拼
- GBK 内码
- 英语-中文
- 可选的代码表输入法
- 简体中文的输入法辅助窗口支持

中文输入法的辅助窗口为所有中文语言环境提供了一个友好并且可以扩展的输入法用户界面。请参见 第 76 页“简体和繁体中文的输入法辅助窗口支持”。

有关中文输入法辅助窗口的更详细信息，请参见《**简体中文用户指南**》和《**繁体中文用户指南**》。

下表显示了 `zh` 语言环境的 TrueType 字体。

表 4–2 `zh_CN.EUC` 语言环境的 TrueType 字体

完整系列名	子系列	格式	供应商	编码
仿宋	R	TrueType	汉仪	GB2312.1980
黑体	R	TrueType	Monotype	GB2312.1980
楷体	R	TrueType	Monotype	GB2312.1980
宋体	R	TrueType	Monotype	GB2312.1980

下表显示 `zh` 语言环境的位图字体。

表 4–3 `zh_CN.EUC` 语言环境的位图字体

完整系列名	子系列	格式	编码
宋体	B	PCF (14,16)	GB2312.1980
宋体	R	PCF (12,14,16,20,24)	GB2312.1980

下表显示 `zh_CN.GBK` 语言环境的 TrueType 字体。

表 4–4 `zh_CN.GBK` 语言环境的 TrueType 字体

完整系列名	子系列	格式	供应商	编码
仿宋	R	TrueType	综艺	GBK
黑体	R	TrueType	综艺	GBK
楷体	R	TrueType	综艺	GBK
宋体	R	TrueType	综艺	GBK

下表显示 `zh_CN.GBK` 语言环境的位图字体。

表 4–5 zh_CN.GBK 语言环境的位图字体

完整系列名	子系列	格式	编码
宋体	R	PCF (12,14,16,20,24)	GBK

下表显示 zh_CN.GB18030 语言环境的 TrueType 字体。

表 4–6 zh_CN.GB18030 语言环境的 TrueType 字体

系列名	子系列	格式	供应商	编码
仿宋	R	TrueType	方正	GB18030–2000
宋体	R	TrueType	方正	GB18030–2000
黑体	R	TrueType	方正	GB18030–2000
楷体	R	TrueType	方正	GB18030–2000

下表显示 zh_CN.GB18030 语言环境的位图字体。

表 4–7 zh_CN.GB18030 语言环境的位图字体

系列名	子系列	格式	编码
宋体	R	PCF (12,14,16,20,24)	GB18030–2000

下表显示受支持的简体中文代码集转换。

表 4–8 简体中文代码集转换

代码	符号	目标代码	符号
GB2312-80	zh_CN.euc	ISO 2022-7	zh_CN.iso2022-7
GB2312-80	zh_CN.euc	ISO 2022-CN	zh_CN.iso2022-CN
GB2312-80	zh_CN.euc	UTF-8	UTF-8
GB18030	zh_CN.gb18030	UTF-8	UTF-8
HZ-GB-2312	HZ-GB-2312	GB2312-80	zh_CN.euc
HZ-GB-2312	HZ-GB-2312	GBK	zh_CN.gbk
HZ-GB-2312	HZ-GB-2312	UTF-8	UTF-8
ISO2022-7	zh_CN.iso2022-7	GB2312-80	zh_CN.euc
ISO2022-CN	zh_CN.iso2022-CN	GB2312-80	zh_CN.euc

表 4–8 简体中文代码集转换　（续下）

代码	符号	目标代码	符号
ISO2022-CN	zh_CN.iso2022-CN	UTF-8	UTF-8
ISO2022-CN	zh_CN.iso2022-CN	zh.GBK	zh_CN.gbk
UTF-8	UTF-8	GB2312-80	zh_CN.euc
UTF-8	UTF-8	GB18030	zh_CN.gb18030
UTF-8	UTF-8	ISO2022-CN	zh_CN.iso2022-CN
UTF-8	UTF-8	zh.GBK	zh_CN.gbk
zh.GBK	zh_CN.gbk	ISO2022-CN	zh_CN.iso2022-CN
zh.GBK	zh_CN.gbk	UTF-8	UTF-8

繁体中文本地化

Solaris 9 产品中的繁体中文提供了五种语言环境：

- zh_TW.EUC 在语言环境中，EUC 方案用于对 CNS11643.1992 代码集进行编码
- zh_TW.BIG5 该语言环境支持 Big5
- zh_TW.UTF-8 该语言环境支持 Unicode 3.1
- zh_HK.BIG5HK 该语言环境支持 Big5-HKSCS
- zh_HK.UTF-8 该语言环境支持 Unicode 3.1

繁体中文主要在中国台湾和香港地区使用。下列输入法在 zh_TW.EUC 、zh_TW.BIG5 和 zh_TW.UTF-8 语言环境中受到支持：

- 新注音
- 注音
- 仓颉
- 行列
- 无虾米
- 大易
- 简易
- 粤语
- EUC 内码
- Big5 内码
- 英语-中文
- 可选代码表输入法（如拼音）
- 繁体中文的输入法辅助窗口支持

下列输入法在 zh_HK.BIG5HK 和 zh_HK.UTF-8 语言环境中受到支持。

- 注音

- 仓颉
- 行列
- 无虾米
- 大易
- 简易
- 粤语
- BIG5+HKSCS 内码
- 英语-中文
- 可选代码表输入法（如拼音）
- 繁体中文的输入法辅助窗口支持
- 新注音

下表显示 zh_TW 语言环境的繁体中文 TrueType 字体。

表 4–9 zh_TW 语言环境的繁体中文 TrueType 字体

完整系列名	子系列	格式	供应商	编码
黑体	R	TrueType	汉仪	CNS11643.1992
楷体	R	TrueType	汉仪	CNS11643.1992
Ming	R	TrueType	汉仪	CNS11643.1992

下表显示 zh_TW 语言环境的繁体中文位图字体。

表 4–10 zh_TW 语言环境的繁体中文位图字体

完整系列名	子系列	格式	编码
Ming	R	PCF (12,14,16,20,24)	CNS11643.1992

下表显示 zh_HK.BIG5HK 语言环境的 TrueType 字体。

表 4–11 zh_HK.BIG5HK 语言环境的 TrueType 字体

系列名	子系列	格式	供应商	编码
Ming	R	TrueType	方正	Big5–HKSCS
黑体	R	TrueType	方正	Big5–HKSCS
楷体	R	TrueType	方正	Big5–HKSCS

下表显示zh_HK.BIG5HK 语言环境的位图字体。

表 4–12 zh_HK.BIG5HK 语言环境的位图字体

系列名	子系列	格式	编码
Ming	R	PCF(12,14,16,20,24)	Big5–HKSCS

下表显示受支持的繁体中文代码集转换。

表 4–13 繁体中文代码集转换

代码	符号	目标代码	符号
BIG5	zh_TW-big5	CNS 11643	zh_TW-euc
BIG5	zh_TW-big5	ISO2022-CN	zh_TW-iso2022-CN-EXT
BIG5	zh_TW-big5	UTF-8	UTF-8
BIG5+HKSCS	zh_HK.big5hk	UTF-8	UTF-8
CNS 11643	zh_TW-euc	BIG5	zh_TW-big5
CNS 11643	zh_TW-euc	UTF-8	UTF-8
CNS 11643	zh_TW-euc	ISO2022-7	zh_TW-iso2022-7
CNS 11643	zh_TW-euc	ISO2022-CN-EXT	zh_TW-iso2022-CN-EXT
CNS 11643	zh_TW-euc	UTF-8	UTF-8
ISO2022-7	zh_TW-iso2022-7	CNS 11643	zh_TW-euc
ISO2022-7	zh_TW-iso2022-7	UTF-8	UTF-8
ISO2022-CN	zh_TW-iso2022-CN-EXT	BIG5	zh_TW-big5
ISO2022-CN-EXT	zh_TW-iso2022-CN-EXT	CNS 11643	zh_TW-euc
UTF-8	UTF-8	BIG5	zh_TW-big5
UTF-8	UTF-8	BIG5+HKSCS	zh_HK.big5hk
UTF-8	UTF-8	CNS 11643	zh_TW-euc
UTF-8	UTF-8	ISO 2022-7	zh_TW-iso2022-7

日语本地化

这部分描述日语语言环境特定的信息。

日语语言环境

在 Solaris 9 环境中，可使用四种支持不同字符集的日语语言环境。`ja` 和 `ja_JP.eucJP` 语言环境基于日语 EUC。`ja_JP.eucJP` 语言环境符合 UI-OSF 日语环境实现协议 1.1 版，`ja` 语言环境符合早期 Solaris 版本中的传统规范。`ja_JP.PCK` 语言环境基于 PC 日文汉字代码（称为 Shift_JIS），而`ja_JP.UTF-8` 基于 `UTF-8` 。

有关日语 EUC 和此字符集之间的映射，请参见 `eucJP`(5) 手册页。 有关 PC 日文汉字代码和此字符集之间的映射，请参见 `PCK(5)` 手册页。

日语字符集

受支持的日语字符集有：

- JIS X 0201–1976
- JIS X 0208–1990
- JIS X 0212–1990
- JIS X 0213–2000（仅限于 Unicode 3.1 中定义的字符）

JIS X 0212–1990 在 `ja_JP.PCK` 语言环境中不受支持。 JIS X 0213–2000 仅在 `ja_JP.UTF-8` 语言环境中受到支持。并不是在 JIS X 0213–2000 中定义的所有字符都可用。只有那些在 Unicode 3.1 字符集中定义的字符可用。

还支持供应商定义的字符 (VDC) 和用户定义的字符 (UDC)。VDC 占用 JIS X 0208–1990 或 JIS X 0212–1990 的未使用（保留）代码点。 除了那些为 VDC 分配的代码点外，UDC 与 VDC 占用相同的代码点。

日语字体

支持三种日语字体格式。位图、TrueType 和 Type1。 日语 Type1 字体只包括用于打印的 JIS X 0212。Type1 字体还由 UDC 使用。

下表对日语位图字体进行了描述。

表 4–14 日语位图字体

完整系列名	子系列	格式	供应商	编码
sun gothic	R、B	PCF(12,14,16,20,24)		JIS X 0208–1983、 JIS X 0201–1976
sun minchou	R	PCF(12,14,16,20,24)		JIS X 0208–1983、 JIS X 0201–1976
ricoh hg gothic b	R	PCF (10,12,14,16,18,20,24)	RICOH	JIS X 0208–1983、JIS X 0201–1976
ricoh hg mincho l	R	PCF (10,12,14,16,18,20,24)	RICOH	JIS X 0208–1983, JIS X 0201–1976
ricoh gothic	R	PCF (10,12,14,16,18,20,24)	RICOH	JIS X 0212–1990、JIS X 0213–2000
ricoh mincho	R	PCF (10,12,14,16,18,20,24)	RICOH	JIS X 0212–1990, JIS X 0213–2000
ricoh heiseimin	R	PCF(12,14,16,18,20,24)	RICOH	JIS X 0212–1990

下表对日语 TrueType 字体进行了描述。

表 4–15 日语 TrueType 字体

完整系列名	子系列	格式	供应商	编码
ricoh hg gothic b	固定	TrueType	RICOH	JIS X 0208–1983、JIS X 0201–1976
ricoh hg mincho l	固定	TrueType	RICOH	JIS X 0208–.1983、JIS X 0201–1976
ricoh gothic	固定、成比例	TrueType	RICOH	JIS X 0201–176、JIS X 0208–1983、JIS X 0213–2000
ricoh mincho	固定、成比例	TrueType	RICOH	JIS X 0201–1976、JIS X 0208–1983、JIS X 0213–2000
ricoh heiseimin	固定	TrueType	RICOH	JIS X 0212–1990

日语输入系统

ATOK12 是 Solaris 9 环境中缺省的 日语输入系统。当安装了日语语言环境后，该输入系统就可用于所有日语语言环境和所有 UTF-8 语言环境。Wnn6 日语输入系统也可用于所有日语语言环境。您可以在“工作区”菜单中切换输入系统。对于日语 Solaris 1.x BCP 支持，可使用 kkcv 日语输入系统。

下面的示例描述如何使用 ATOK12 进行日语输入。

1. 按 Control + spacebar 打开转换模式。
2. 键入假名字符文本（例如kanjihenkan）。
3. 按空格键转换为（日语）汉字字符。

   要显示其它（日语）汉字字符，请按空格键显示转换候选表。键入您要选择的数字。
4. 要将整个文本转换为（日语）汉字字符文本，请按回车键。

   按向下箭头键只转换选定的字符。
5. 按 Control + spacebar 关闭转换模式。

日语终端的终端设置

在基于字符的终端 (TTY) 上使用日语语言环境要求您使用终端设置以便使行编辑正确工作。

- 如果您的终端是 CDE 终端仿真器 (dtterm)，请在任何日语语言环境（ja、ja_JP.PCK 或 ja_JP.UTF-8）中使用带参数 -defeucw 的 stty(1)。ja 语言环境中的一个示例是：

  ```
  % setenv LANG ja
  % stty defeucw
  ```
- 如果您的终端不是 CDE 终端仿真器，但终端的代码集与当前语言环境的代码集相同，请使用带参数 -defeucw 的 stty(1)。
- 如果您的终端的代码集与当前语言环境的代码集不匹配，请使用 setterm(1) 以启用代码转换。例如，如果您正在使用语言环境 ja，但您的终端要求 PCK（Shift_JIS 代码），则请指定：

  ```
  % setenv LANG ja
  % setterm -x PCK
  ```
  有关详细信息，请参见 setterm(3CURSES) 手册页。

日语 iconv 模块

iconv(1) 和 iconv(3) 支持若干种日语代码集转换。 有关详细信息，请参见 iconv_ja(5) 手册页。

用户定义的字符支持

用户定义的字符公用程序 sdtudctool 处理轮廓 (Type1) 和位图 (PCF) 字体。 某些公用程序也可用于迁移由早期版本中的旧公用程序创建的 UDC 字体，如 fontedit, type3creator 和 fontmanager。

不完全语言环境和完全语言环境之间的差别

下列组件只在带有 Language CD 的日语 完全语言环境中可用：

- 已翻译的信息、帮助和手册页
- Wnn6 日语输入系统
- 日语 Solaris 1.x BCP 支持
- Mincho (min*) 字样位图字体
- 用于打印的 JIS X 0212 Type1 字体
- 日语特定的哑打印机和 jpostprint 支持
- 传统的日语公用程序，如 kanji(1)

朝鲜语本地化

1995 年 12 月，韩国政府公布了标准朝鲜语代码集 KS X 1005–1，它基于 ISO 10646-1/Unicode 2.0。

ISO-10646 字符集使用了两种通用字符集：

- UCS-2。通用字符集（双字节形式）
- UCS-4。通用字符集（四字节形式）

ISO-10646 字符集不能直接在基于 IBM PC 的操作系统上使用。例如，Solaris 操作环境的内核和许多其它模块将某些字节值翻译为控制指令，如所有字符串中的空字符 (0x00)。ISO-10646 字符集可以使用第一个或随后字节中的任何位组合进行编码。在存在这些限制的情况下，ISO-10646 字符不能在 Solaris 系统中自由传输。

为了建立迁移路径，ISO-10646 字符集定义了 UCS 变换格式 (UTF)，它无须使用 C0 控制 (0x00..0x1F)、C1 控制 (0x80..0x9F)、空格 (0x20) 和 DEL (0x7F) 即可对 ISO-10646 字符进行重新编码。

ko.UTF-8 是一个用以支持朝鲜语标准代码集 KS X 1005–1 的 Solaris 语言环境。该语言环境支持以前的 KS X 1005 中的所有字符和全部 11172 个朝鲜语字符。朝鲜语 UTF-8 支持与朝鲜语相关的 ISO-10646 字符和字体。由于 ISO-10646 包括了世界上的所有字符，因而提供了所有的各式各样的输入法和字体，以便您可以输入和输出任何语言的任何字符。在可以使用通用的 UTF/UCS 之前，朝鲜语 UTF-8 支持与朝鲜语字符以及以前的朝鲜语标准代码集中的所有其它字符有关的 ISO-10646 代码子集，并支持扩展 ASCII。

在 ko 语言环境中，EUC 方案用于对 KS X 1001 进行编码。 ko.UTF-8 语言环境支持 KS X 1005–1/Unicode 3.1 代码集，它是 KS X 1001 的超集。 这两种语言环境在最终用户看来是一样的，但它们的内部字符编码并不相同。朝鲜语 Solaris 产品支持下列输入法：

对于 ko 语言环境：

- 朝鲜语文字 2–BeolSik（一个辅音集和一个元音集）

- 朝鲜语文字/朝鲜语汉字转换
- 特殊字符
- 十六进制代码

对于 ko.UTF-8 语言环境：

- 朝鲜语文字 2–BeolSik（一个辅音集和一个元音集）
- 朝鲜语文字/朝鲜语汉字转换
- 特殊字符
- 十六进制代码

下表显示 ko 语言环境的朝鲜语位图字体。

表 4–16 ko 语言环境的 Solaris 9 朝鲜语位图字体

完整系列名	子系列	格式	编码
Gothic	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Graphic	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Haeso	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Kodig	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Myeongijo	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Pilki	R/B	PCF (12,14,16,18,20,24)	KS X 1001
Round gothic	R/B	PCF (12,14,16,18,20,24)	KS X 1001

下表显示 ko.UTF-8 语言环境的朝鲜语位图字体。

表 4–17 ko.UTF-8 语言环境的 Solaris 9 朝鲜语位图字体

完整系列名	子系列	格式	编码
Gothic	R/B	PCF (12,14,16,18,20,24)	KS X 1001（Johap）
Graphic	R/B	PCF (12,14,16,18,20,24)	KS X 1001（Johap）
Haeso	R/B	PCF (12,14,16,18,20,24)	KS X 1001（Johap）
Kodig	R/B	PCF (12,14,16,18,20,24)	KS X 1001（Johap）
Myeongijo	R/B	PCF (12,14,16,18,20,24)	KS X 1001（Johap）
Pilki	R/B	PCF (12,14,16,18,20,24)	KS X 1001（Johap）

下表显示 ko/ko.UTF-8 语言环境的朝鲜语 TrueType 字体。

表 4–18 ko/ko.UTF-8 语言环境的 Solaris 9 朝鲜语 TrueType 字体

完整系列名	子系列	格式	供应商	编码
Kodig/Gothic	R	TrueType	Hanyang	Unicode
Myeongijo	R	TrueType	Hanyang	Unicode
Haeso	R	TrueType	Hanyang	Unicode
Round gothic	R	TrueType	Hanyang	Unicode

下表显示朝鲜语 iconv。

表 4–19 朝鲜语 iconv

代码	符号	目标代码	符号
IBM CP933	cp933	UTF-8 (Unicode 2.0)	ko_KR-UTF-8
ISO646	646	KS X 1001	5601
ISO2022-KR	iso2022-7	KS X 1001	ko_KR-euc
ISO2022-KR	iso2022-7	UTF-8 (Unicode 2.0)	ko_KR-UTF-8
KS X 1001	5601	UTF-8	UTF-8
KS X 1001	EUC-KR	UTF-8	UTF-8
KS X 1001	KSC5601	UTF-8	UTF-8
KS X 1001	ko_KR-euc	UTF-8 (Unicode 2.0)	ko_KR-UTF-8
KS X 1001	ko_KR-euc	ISO2022-KR	ko_KR-iso2022-7
KS X 1001	ko_KR-euc	KS X 1001	ko_KR-johap
KS X 1001	ko_KR-euc	KS X 1001	ko_KR-johap92
KS X 1001	ko_KR-euc	KS X 1001	ko_KR-nbyte
KS X 1001	ko-KR-nbyte	KS X 1001	ko_KR-euc
KS X 1001	ko-KR-johap	UTF-8 (Unicode 2.0)	ko_KR-UTF-8
KS X 1001	ko-KR-johap	KS X 1001	ko_KR-euc
KS X 1001	ko-KR-johap92	UTF-8 (Unicode 2.0)	ko_KR-UTF-8
KS X 1001	ko-KR-johap92	KS X 1001	ko_KR-euc
UTF-8	UTF-8	KS X 1001	5601
UTF-8	UTF-8	KS X 1001	EUC-KR
UTF-8	UTF-8	KS X 1001	KSC5601

表 4–19 朝鲜语 iconv （续下）

代码	符号	目标代码	符号
UTF-8	ko-KR-UTF-8	IBM CP 933	cp 933
UTF-8	ko-KR-UTF-8	KS X 1001	ko_KR-euc
UTF-8	ko-KR-UTF-8	ISO2022-KR	ko_KR-iso2022-7
UTF-8	ko-KR-UTF-8	KS X 1001	ko_KR-johap
UTF-8	ko-KR-UTF-8	KS X 1001	ko_KR-johap92

第 5 章

UTF-8 语言环境支持概述

本部分描述以下内容：

- 第 91 页 "Unicode 概述"
- 第 92 页 "Unicode 语言环境：en_US.UTF-8 支持概述"
- 第 94 页 "桌面输入法"
- 第 111 页 "系统环境"
- 第 115 页 "代码转换"
- 第 116 页 "DtMail"
- 第 118 页 "程序设计环境"

Unicode 概述

Unicode 标准是用于表示文本以供计算机进行处理的通用字符编码标准。 它与国际标准 ISO/IEC 10646-1:2000 和 ISO/IEC 10646–2:2001 完全兼容，并包含与 ISO/IEC 10646 相同的全部字符和编码点。Unicode 标准提供有关各字符及其用法的附加信息。符合 Unicode 的所有执行程序也同时符合 ISO/IEC 10646。

Unicode 标准提供了一种对多语种纯文本进行一致编码的方法，便于国际文本文件的交换。处理多语种文本的计算机用户 、商务人员、语言学家、研究人员、科学家等会发现 Unicode 标准可大大简化他们的工作。Unicode 标准对于经常使用数学符号和其它技术字符的数学家和技术人员也很有价值。

Unicode 能支持的最大可能代码数据点数是 1,114,112，位于 17 个 16 位平面中。每一个平面都可支持 65,536 个不同的代码数据点。

在 Unicode 可支持的一百多万个代码数据点中，3.1 版目前定义了 94,140 个字符，它们位于平面 0、1、2 和 14 中。 平面 15 和 16 供个人使用，也称为用户定义的字符。平面 15 和 16 总共可支持 131,068 个用户定义的字符。

Unicode 可使用下列任何一种字符编码方案来编码：

- UTF-8
- UTF-16
- UTF-32

UTF-8 是 Unicode 的一种可变长度编码形式，它透明地保留了 ASCII 字符代码值。该形式在 Solaris Unicode 语言环境中用作文件代码。

UTF-16 是 Unicode 的一种 16 位编码形式。在 UTF-16 中，多达 65,535 个字符被编码为单个 16 位值。映射在 65,535 到 1,114,111 的字符被编码为成对的 16 位值（代理）。

UTF-32 是 Unicode 的一种固定长度的 21 位编码形式，通常用在 32 位容器或数据类型中。该形式在 Solaris Unicode 语言环境中用作进程代码（宽字符代码）。

有关 Unicode 标准和 ISO/IEC 10646 及其各种表示形式的详细信息，请参考：

- *Unicode* **标准** *3.0* **版**、*Unicode* **标准附件** *#19*： *UFT-32* 和 *Unicode* **标准附件** *#27*： *Version 3.1*（来自 Unicode 协会）
- ISO/IEC 10646-1:2000、信息技术-通用多八进制字符集 (UCS) - 第 1 部分： 体系结构和基本多语种平面
- ISO/IEC 10646-2： 信息技术-通用多八进制字符集 (UCS) - 第 2 部分： 脚本和符号的次要多语种平面、CJK 象形文字的辅助平面、特殊用途平面
- Unicode Consortium web 站点： http://www.unicode.org/。

Unicode 语言环境：`en_US.UTF-8` 支持概述

Unicode/UTF-8 语言环境支持 Unicode 3.1。 `en_US.UTF-8` 语言环境通过将 `UTF-8` 用作其代码集提供多脚本处理支持。该语言环境处理多种语言中的输入和输出文本，是 Solaris 操作环境中第一个具有此功能的语言环境。其它 UTF-8 语言环境的功能类似于 `en_us.UTF-8` 的功能； 后面对 `en_US.UTF-8` 的探讨同样适用于这些语言环境。

注意：`UTF-8` 是由 X/Open-Uniforum Joint Internationalization Working Group (XoJIG) 在 1992 年制定的 Unicode / ISO/IEC 10646-1 的文件系统安全通用字符集转换格式，并在 1996 年作为 ISO/IEC 10646-1:1993 的第二修正案被 ISO 和 IEC 批准。该标准已由 Unicode Consortium、国际标准化组织和国际电工技术委员会采用而成为 Unicode 2.0 和 ISO/IEC 10646-1 的一部分。

Solaris 中的 Unicode 语言环境支持对在 Unicode 3.1 与 ISO/IEC 10646-1 和 10646-2 中定义的每一个代码数据点值的处理。 所支持的语言不仅包括全欧洲语言和亚洲语言，还包括一些复杂的文本布局语言，如阿拉伯语、希伯来语、印地文和泰语。由于字体资源有限，Solaris 9 软件仅包括来自下列字符集的字符字型：

- ISO 8859-1（大多数西欧语言，如英语、法语、西班牙语和德语）
- ISO 8859-2（大多数中欧语言，如捷克语、波兰语和匈牙利语）
- ISO 8859-4（斯堪的纳维亚和波罗的海语）
- ISO 8859-5（俄语）
- ISO 8859-6（阿拉伯语，包括许多更多的表示形式字符字型）
- ISO 8859–7（希腊语）
- ISO 8859–8（希伯来语）
- ISO 8859-9（土耳其语）
- TIS 620.2533（泰语，包括许多更多的表示形式字符字型）
- ISO 8859–15（大多数带有欧元符号的西欧语言）
- GB 2312–1980（简体中文）
- JIS X 0201–1976, JIS X 0208–1990（日语）
- KSC 5601–1992 附件 3（朝鲜语）
- GB 18030（简体中文）
- HKSCS（繁体中文，中国香港特别行政区）
- Big5（繁体中文，中国台湾地区）
- IS 13194.1991，也称为 ISCII（印地文，包括许多更多的表示形式字符字型）

如果您试图查看 `en_US.UTF-8` 语言环境中没有其相应字型的字符，则该语言环境将显示“no-glyph”字型，如下所示：



语言环境可以在安装时选择，并可指定为系统的缺省语言环境。

对于 64 位和 32 位的 Solaris 系统，提供了同一级别的 `en_US.UTF-8` 语言环境支持。

注意：Motif 和 CDE 桌面应用程序和库支持 `en_US.UTF-8` 语言环境。但 XView™ 和 OLIT 库**不** 支持 `en_US.UTF-8` 语言环境。

桌面输入法

CDE 提供使用 Xm Toolkit 为国际化的应用程序输入本地化输入的能力。启用了 `XmText[Field]` 小部件以便与各个语言环境中的输入法相连接。输入法被国际化，因为某些语言环境以从右向左、从上到下等方向写入它们的文本。在同一个应用程序中，您可以使用应用多种字体的不同输入法。

预编辑区显示正被预编辑的字符串。这可以通过四种方式来完成：

- OffTheSpot
- OverTheSpot（缺省）
- Root
- None

在 OffTheSpot 模式下，此位置在状态区右边主窗口的正下方。在 OverTheSpot 模式下，预编辑区位于光标位置处。在 Root 模式下，预编辑区和状态区与客户程序窗口分离。

有关详细信息，请参考 `VendorShell`(3X) 手册页上的 `XmNpreeditType` 资源描述。

注意：在 Solaris 9 环境中，对简体/繁体中文、日语和朝鲜语可采用本地亚洲语言输入法。这些方法是对 Unicode 语言环境的当前多脚本输入法的补充。

下面的讨论包括对所选的输入法、如何使用这些输入法以及如何在这些方法之间切换的描述。

语言选择和输入模式

Solaris Unicode 语言环境支持多种语言。每个 Unicode 语言环境共有 14 种输入模式：

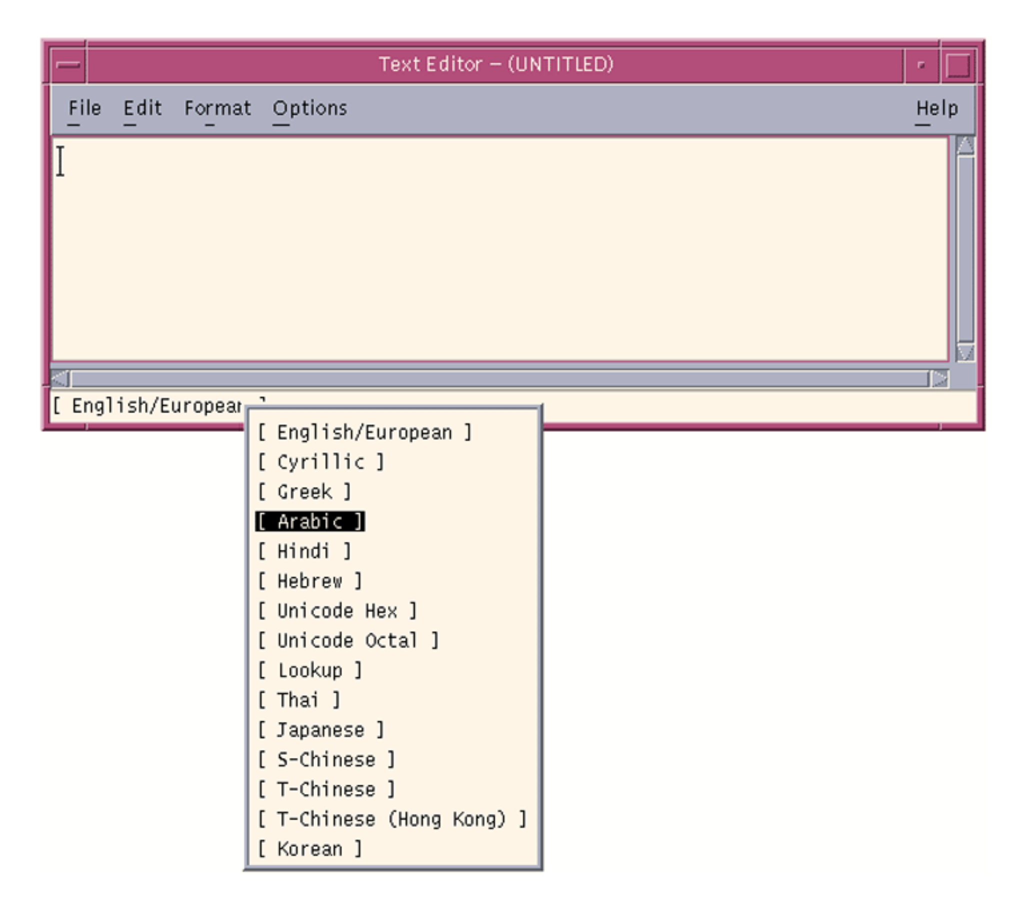
- 英语/欧洲语言

- 西里尔文
- 希腊文
- 阿拉伯文
- 希伯来文
- 泰文
- 日语
- 朝鲜语
- 简体中文
- 繁体中文
- 繁体中文（香港特别行政区）
- 印地文

- Unicode 十六进制和八进制代码输入法
- 查表输入法

访问输入法

您可以使用书写键组合或输入模式选择窗口来切换到某个特定的输入模式。要访问输入模式选择窗口，请在应用程序窗口左下角的状态区中按下鼠标按钮 1。输入模式选择窗口显示在下图中。



图形 5–1 输入模式选择窗口

输入模式切换键序列

您可以使用 表 5–1 中列出的键序列将当前输入模式更改为新的输入模式。 使用这些键序列的唯一限制是，如果您当前正使用某个亚洲语言输入模式，则需要同时按下 Control 键和空格键切换回英语/欧洲语言输入模式。一旦处于英语/欧洲语言输入模式，您就可以使用这些键序列随意切换到其它任何一种输入模式中。

下列键序列显示如何从英语/欧洲语言输入模式切换到西里尔文输入模式：

- 按下书写键。
- 按下并释放 c。
- 按下 c。

表 5–1 输入模式切换键序列

键序列	输入模式
Control + Spacebar	英语/欧洲语言
Compose c c	西里尔文
Compose g g	希腊文
Compose a r	阿拉伯文
Compose h h	希伯来文
Compose t t	泰文
Compose h i	印地文
Compose j a	日语
Compose k o	朝鲜语
Compose s c	简体中文
Compose t c	繁体中文
Compose h k	繁体中文（香港特别行政区）
Compose u o	Unicode 八进制代码输入法
Compose u h	Unicode 十六进制代码输入法
Compose l l	查表输入法

英语/欧洲语言输入模式

英语/欧洲语言输入模式包括英语字母和带有变音标记的字符（例如 á、è、î、õ 和 ü）以及欧洲语言中的一些字符（如 ¡, §, ¿）。

该输入模式是所有应用程序的缺省模式。 输入模式显示在 GUI 应用程序窗口的左下角。

要插入带有变音标记的字符或 Latin-1、Latin-2、Latin-4、Latin-5 和 Latin-9 中的特殊字符，必须键入一个书写键序列，如下例所述。

要显示 Ä 字符：

- 按下并释放书写键。
- 同时按下 Shift 和 A 键。释放 Shift 和 A 键。

■ 按下并释放 " 键。

要显示 ¿, 字符：

■ 按下并释放书写键。
■ 按下并释放 ? 键。
■ 按下并释放 ? 键。

如果键盘上没有书写键，您可以模仿书写键的操作，方法是同时按下 Control 键和 Shift 键。

要从该语言环境中输入欧元货币符号（Unicode 值为 U+20AC），则可以使用下列任何一个输入序列：

■ 同时按下 AltGraph 和 E
■ 同时按下 AltGraph 和 4
■ 同时按下 AltGraph 和 5

对于这些输入序列，您可以同时按下这两个键。如果键盘上没有 AltGraph 键，则您可以用 Alt 键来替代。

下面各表显示 Solaris 环境中最常用的 Latin-1、Latin-2、Latin-3、Latin-4、Latin-5 和 Latin-9 语言输入的书写键。

下表列出常用的 Latin-1 书写键序列。

表 5–2 常用的 Latin-1 书写键序列

按下 Compose，然后按下并释放	然后按下并释放	结果
spacebar	spacebar	无间断间隔
s	1	上标 1
s	2	上标 2
s	3	上标 3
!	!	竖翻叹号
x	o	货币符号 ¤
p	!	段落符号 ¶
/	u	mu u
'	"	锐音符 ´
,	,（逗号）	下加符 Ç
"	"	分音符 ¨
-	^	长音符 ¯

表 5–2 常用的 Latin-1 书写键序列 （续下）

按下 Compose，然后按下并释放	然后按下并释放	结果
o	o	度 °
x	x	乘号 x
+	-	正负号 ±
-	-	软连字符 –
-	:	除号 ÷
-	a	序号（阴性）ª
-	o	序号（阳性）º
-	,（逗号）	非标记 ¬
.	.	中间点 ·
1	2	普通分数 ½
1	4	普通分数 ¼
3	4	普通分数 ¾
<	<	左双尖引号 «
>	>	右双尖引号 »
?	?	竖翻问号 ¿
A	`（反引号）	带抑音符的 A À
A	'（单引号）	带锐音符的 A Á
A	*	带上圆圈的 A Å
A	"	带分音符的 A Ä
A	^	带扬抑符的 A Â
A	~	带波浪号的 A Ã
A	E	双元音 AE Æ
C	,（逗号）	带下加符的 C Ç
C	o	版权所有标记 ©
D	-	大写 eth ð
E	`（反引号）	带抑音符的 E È
E	'	带锐音符的 E É
E	"	带分音符的 E Ë

表 5–2 常用的 Latin-1 书写键序列 （续下）

按下 Compose，然后按下并释放	然后按下并释放	结果
E	^	带扬抑符的 E Ê
I	`（反引号）	带抑音符的 I Ì
I	'	带锐音符的 I Í
I	"	带分音符的 I Ï
I	^	带扬抑符的 I Î
L	-	英镑标记 £
N	~	带波浪号的 N Ñ
O	`（反引号）	带抑音符的 O Ò
O	'	带锐音符的 O Ó
O	/	带斜杠的 O Ø
O	"	带分音符的 O Ö
O	^	带扬抑符的 O Ô
O	~	带波浪号的 O Õ
R	O	注册商标标记 ®
T	H	拉丁文字母 Thorn þ
U	`（反引号）	带抑音符的 U Ù
U	'	带锐音符的 U Ú
U	"	带分音符的 U Ü
U	^	带扬抑符的 U Û
Y	'	带锐音符的 Y ý
Y	-	日元标记 ¥
a	`（反引号）	带抑音符的 a à
a	'	带锐音符的 a á
a	*	带上圆圈的 a å
a	"	带分音符的 a ä
a	~	带波浪号的 a ã
a	^	带扬抑符的 a â
a	e	双元音 ae æ

表 5–2 常用的 Latin-1 书写键序列 *（续下）*

按下 `Compose`，然后按下并释放	然后按下并释放	结果
`c`	`,`（逗号）	带下加符的 c ç
`c`	`/`	百分号 ¢
`c`	`o`	版权所有标记 ©
`d`	`-`	eth ð
`e`	`` ` ``（反引号）	带抑音符的 e è
`e`	`'`	带锐音符的 e é
`e`	`"`	带分音符的 e ë
`e`	`^`	带扬抑符的 e ê
`i`	`` ` ``（反引号）	带抑音符的 i ì
`i`	`'`	带锐音符的 i í
`i`	`"`	带分音符的 i ï
`i`	`^`	带扬抑符的 i î
`n`	`~`	带波浪号的 n ñ
`o`	`` ` ``（反引号）	带抑音符的 o ò
`o`	`'`	带锐音符的 o ó
`o`	`/`	带斜杠的 o ø
`o`	`"`	带分音符的 o ö
`o`	`^`	带扬抑符的 o ô
`o`	`~`	带波浪号的 o õ
`s`	`s`	德语双 s ß，也称为清音 S
`t`	`h`	拉丁文字母 Thorn þ
`u`	`` ` ``（反引号）	带抑音符的 u ù
`u`	`'`	带锐音符的 u ú
`u`	`"`	带分音符的 u ü
`u`	`^`	带扬抑符的 u û
`y`	`'`	带锐音符的 y y
`y`	`"`	带分音符的 y ÿ
|	|	间断条 ¦

下表列出通用的 Latin-2 和 Latin-4 书写键序列。

表 5–3 通用的 Latin-2 书写键序列

按下 Compose，然后按下并释放	按下并释放	结果
a	spacebar	ogonek
u	spacebar	短音符
v	spacebar	抑扬符
"	spacebar	双锐音符
A	a	A Ogonek
A	u	带短音符的 A
C	'	带锐音符的 C
C	v	带抑扬符的 C
D	v	带抑扬符的 D
-	D	带斜线的 D
E	v	带抑扬符的 E
E	a	E Ogonek
L	'	带锐音符的 L
L	-	带斜线的 L
L	>	带抑扬符的 L
N	'	带锐音符的 N
N	v	带抑扬符的 N
O	>	带双锐音符的 O
S	'	带锐音符的 S
S	v	带抑扬符的 S
S	,	带下加符的 S
R	'	带锐音符的 R
R	v	带抑扬符的 R
T	v	带抑扬符的 T
T	,	带下加符的 T
U	*	带上圆圈的 U
U	>	带双锐音符的 U

表 5–3 通用的 Latin-2 书写键序列　　*（续下）*

按下 Compose，然后按下并释放	按下并释放	结果
Z	'	带锐音符的 Z
Z	v	带抑扬符的 Z
Z	.	带上点的 Z
k	k	拉丁文字母 kra
A	_	带长音符的 A
E	_	带长音符的 E
E	.	带上点的 E
G	,	带下加符的 G
I	_	带长音符的 I
I	~	带波浪号的 I
I	a	I Ogonek
K	,	带下加符的 K
L	,	带下加符的 L
N	,	带下加符的 N
O	_	带长音符的 O
R	,	带下加符的 R
T	|	带斜线的 T
U	~	带波浪号的 U
U	a	U Ogonek
U	_	带长音符的 U
N	N	拉丁文字母 Eng
a	_	带长音符的 a
e	_	带长音符的 e
e	.	带上点的 e
g	,	带下加符的 g
i	_	带长音符的 i
i	~	带波浪号的 i
i	a	i Ogonek

表 5–3 通用的 Latin-2 书写键序列 *（续下）*

按下 Compose，然后按下并释放	按下并释放	结果
k	,	带下加符的 k
l	,	带下加符的 l
n	,	带下加符的 n
o	_	带长音符的 o
r	,	带下加符的 r
t	|	带斜线的 t
u	~	带波浪号的 u
u	a	u Ogonek
u	_	带长音符的 u
n	n	拉丁文字母 Eng

下表列出通用的 Latin-3 书写键序列。

表 5–4 通用的 Latin-3 书写键序列

按下 Compose，然后按下并释放	按下并释放	结果
C	>	带扬抑符的 C
C	.	带上点的 C
G	>	带扬抑符的 G
G	.	带上点的 G
H	>	带扬抑符的 H
J	>	带扬抑符的 j
S	>	带扬抑符的 S
U	u	带短音符的 U
c	>	带扬抑符的 c
c	.	带上点的 c
g	>	带扬抑符的 g
g	.	带上点的 g
h	>	带扬抑符的 h

表 5–4 通用的 Latin-3 书写键序列 （续下）

按下 Compose，然后按下并释放	按下并释放	结果
j	>	带扬抑符的 j
s	>	带扬抑符的 s
u	u	带短音符的 u

下表列出通用的 Latin-5 书写键序列。

表 5–5 通用的 Latin-5 书写键序列

按下 Compose，然后按下并释放	按下并释放	结果
G	u	带短音符的 G
I	.	带上点的 I
g	u	带短音符的 g
i	.	无点的 i

下表列出通用的 Latin-9 书写键序列。

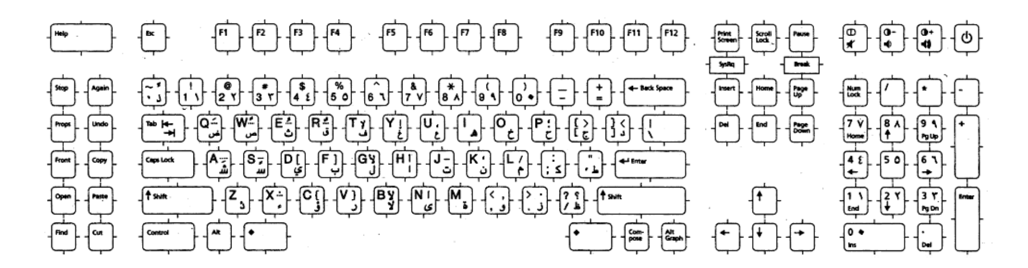
表 5–6 通用的 Latin-9 书写键序列

按下 Compose，然后按下并释放	按下并释放	结果
o	e	连字 oe
O	E	连字 OE
Y	"	带分音符的 Y

阿拉伯文输入模式

要切换到阿拉伯文输入模式，请键入 Compose a r，或者请从输入模式选择窗口中选择阿拉伯文。有关访问输入模式选择窗口的信息，请参见 图形 5–1。

下图显示阿拉伯文键盘布局。

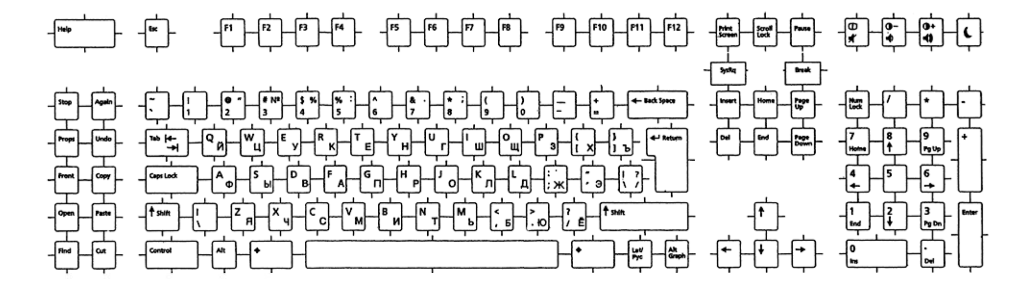


图形 5–2 阿拉伯文键盘

西里尔文输入模式

要切换到西里尔文输入模式，请按下 Compose c c ，或者请从输入模式选择窗口中选择西里尔文。 有关访问输入模式选择窗口的信息，请参见 图形 5–1。

下图显示西里尔文（俄文）键盘布局。



图形 5–3 西里尔文（俄文）键盘

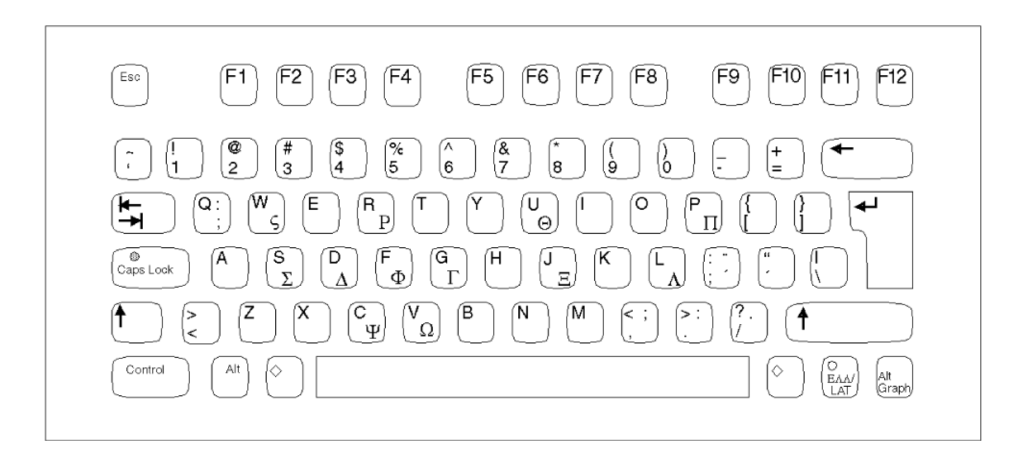
在切换到西里尔文输入模式后，您不能输入英语或欧洲语言文本。要切换回英语/欧洲语言输入模式，请同时键入 Control 和 spacebar，或者通过单击状态区从输入模式选择窗口中选择英语/欧洲语言输入模式。请参见 图形 5–1。

您也可以通过键入相应的输入模式切换键序列切换到其它输入模式中。

希腊文输入模式

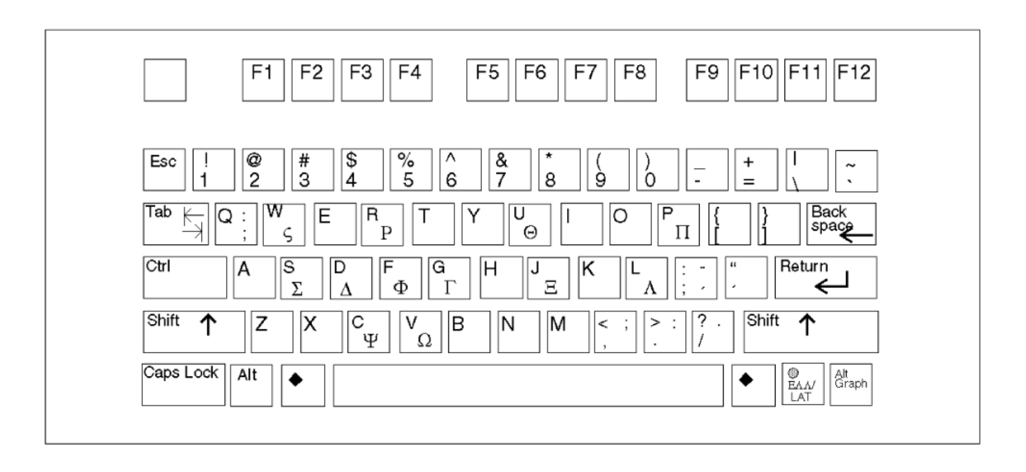
要切换到希腊文输入模式，请按下 Compose g g，或者从输入模式选择窗口中选择希腊文。有关访问输入模式选择窗口的信息，请参见图形 5–1。

在切换到希腊文输入模式后，您不能输入英语或欧洲语言文本。要切换回英语/欧洲语言输入模式，请同时键入 Control 和 spacebar，或者通过单击状态区从输入模式选择窗口中选择英语/欧洲语言输入模式。下图显示希腊文欧元键盘布局。



图形 5–4 希腊文欧元键盘

下图显示希腊文 UNIX 键盘。

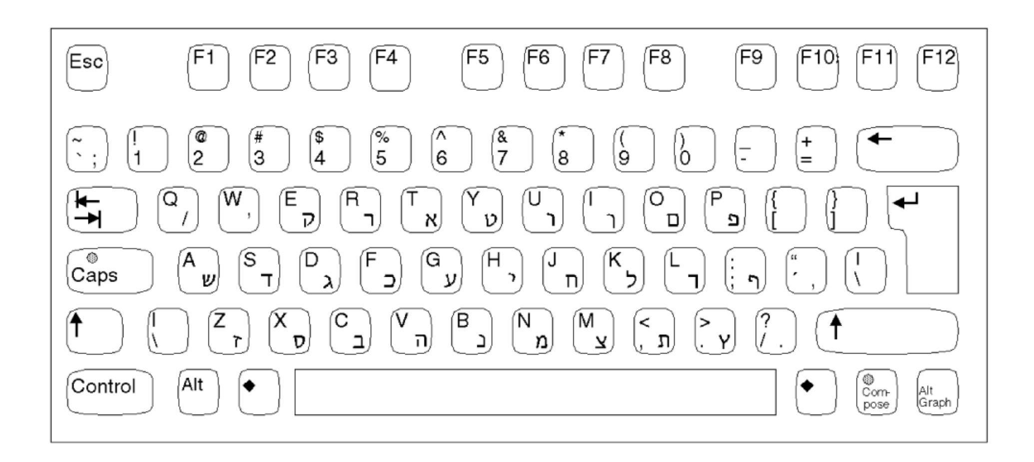


图形 5–5 希腊文 UNIX 键盘

希伯来文输入模式

要切换到希伯来文输入模式，请按下 `Compose h h`，或者从输入模式选择窗口中选择希伯来文。 有关访问输入模式选择窗口的信息，请参见图形 5–1.

下图显示希伯来文键盘布局。



图形 5–6 希伯来文键盘

印地文输入模式

要切换到印地文输入模式，请按下 `Compose h i`，或者从输入模式选择窗口中选择印地文。有关访问输入模式选择窗口的详细信息，请参见 图形 5–1。要切换回英语/欧洲语言输入模式，请同时键入 `Control` 和 `spacebar`，或者通过单击状态区从输入模式选择窗口中选择英语/欧洲语言输入模式。



图形 5–7 印地文键盘

下图显示印地文 Shift 键盘。

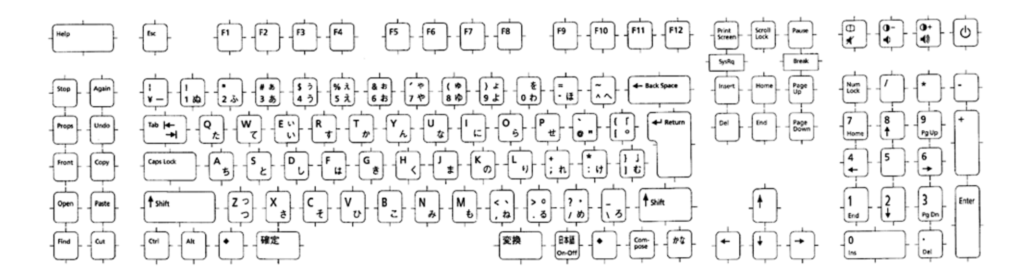


图形 5–8 印地文 Shift 键盘

日语输入模式

要切换到日语输入模式，请按下 `Compose j a` ，或者从输入模式选择窗口中选择日语。有关访问输入模式选择窗口的信息，请参见图形 5–1。

要使用本地日语输入系统，需要安装一个或多个日语语言环境，然后重新引导系统。安装了日语语言环境后，您可以在所有 `UTF-8` 语言环境中使用 ATOK12。 Wnn6 在 `UTF-8` 语言环境中不可用（`ja_JP.UTF-8` 除外）。

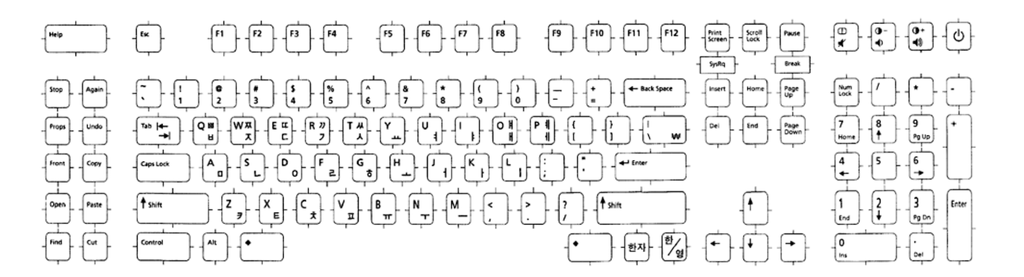


图形 5–9 日语键盘

朝鲜语输入模式

要切换到朝鲜语输入模式，请按下 `Compose k o`，或者从输入模式选择窗口中选择朝鲜语。有关访问输入模式选择窗口的信息，请参见 图形 5–1。

要拥有本地朝鲜语输入系统，需要在系统上安装一个或多个朝鲜语语言环境。一旦安装了朝鲜语语言环境，您就可以使用本地朝鲜语输入系统了。有关如何使用朝鲜语输入系统的详细信息，请参考《*Korean Solaris User's Guide*》。



图形 5–10 朝鲜语键盘

简体中文输入模式

要切换到简体中文输入模式，请按下 `Compose s c`，或者从输入模式选择窗口中选择简体中文。有关访问输入模式选择窗口的信息，请参见图形 5–1。

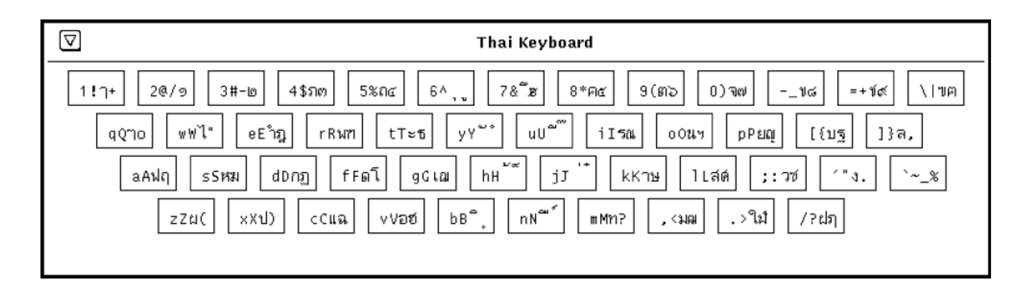
要使用本地简体中文输入系统，需要在系统上安装一个或多个简体中文语言环境。有关如何使用简体中文输入系统的详细信息，请参考《**简体中文 *Solaris* 用户指南**》。

泰语输入模式

要切换到泰语输入模式，请按下 `Compose t t`，或者从输入模式选择窗口中选择泰语。有关访问输入模式选择窗口的详细信息，请参见 图形 5–1。

要切换回英语/欧洲语言输入模式，请同时键入 `Control` 和 `spacebar`，或者通过单击状态区从输入模式选择窗口中选择英语/欧洲语言输入模式。

下图显示泰语键盘布局。



图形 5–11 泰语键盘

繁体中文输入模式

要切换到繁体中文输入模式，请按下 `Compose t c`，或者从输入模式选择窗口中选择繁体中文。有关访问输入模式选择窗口的信息，请参见 图形 5–1。

要访问本地繁体中文输入系统，需要在系统上安装一个或多个繁体中文语言环境。有关如何使用繁体中文输入系统的详细信息，请参考《*Traditional Chinese Solaris User's Guide*》。

繁体中文（中国香港特别行政区）输入模式

要切换到繁体中文输入模式，请按下 `Compose h k`，或者从输入模式选择窗口中选择繁体中文（中国香港特别行政区）。有关访问输入模式选择窗口的信息，请参见 图形 5–1。

要访问本地繁体中文（中国香港特别行政区）输入系统，需要在系统上安装一个或多个繁体中文（中国香港特别行政区）语言环境。

Unicode 十六进制输入模式

要切换到 Unicode 十六进制代码输入法输入模式，请按下 `Compose u h`，或者从输入模式选择窗口中选择 Unicode 十六进制。要切换到八进制数字系统，请按下 `Compose u o` 或者选择 Unicode 八进制。有关访问输入模式选择窗口的信息，请参见图形 5–1。

要使用这些输入模式，您需要了解字符的十六进制或八进制代码数据点值。有关代码数据点值和字符之间的映射的信息，请参考《*Unicode* **标准** *3.0* **版**》。

如果您正处于 Unicode 十六进制代码输入模式，则要输入一个字符，需要键入四位十六进制数字。以下是一些十六进制值样例：

- 00A1 表示竖翻叹号

- 03B2 表示希腊文小写字母 Beta
- AC00 表示朝鲜语 Hangul 音节
- 30A1 表示日语片假名字母 A
- 4E58 表示统一汉字

您可以使用 A、B、C、D、E 和 F 的大写和小写字母形式来表示十六进制数字。如果您首选使用八进制数字系统而不是十六进制数字，则可以输入八进制数字 0 到 7。 如果一位或两位数字键入错误，则可以使用 `Delete` 或 `Backspace` 键删除这些数字。

查表输入模式

要切换到查表输入模式，请按下 `Compose l l`，或者从输入模式选择窗口中选择查找。有关访问输入模式选择窗口的信息，请参见图形 5–1。

第二个查找窗口显示仅限于成组显示的候选项，一次最多显示 80 个候选项。按下 `Control n` 转到下一组候选项，或者按下 `Control p` 转到上一组候选项。

系统环境

本部分描述语言环境环境变量、TTY 环境设置、32 位和 64 位 STREAMS 模块以及终端支持。

语言环境环境变量

确保您的系统已安装了 `en_US.UTF-8` 语言环境。要检查各种类中的当前语言环境设置，请使用 `locale`(1) 公用程序。

```
system% locale
LANG=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_ALL=
```

要使用 `en_US.UTF-8` 语言环境桌面环境，请首先选择该语言环境。在 TTY 环境中，请首先选择语言环境，方法是将 `LANG` 环境变量设置为 `en_US.UTF-8` ， 如下面的 C-shell 示例所示：

```
system% setenv LANG en_US.UTF-8
```

确保 LC_ALL、 LC_COLLATE、 LC_CTYPE、LC_MESSAGES、LC_NUMERIC、LC_MONETARY 和 LC_TIME 种类未设置，或者设置为 en_US.UTF-8。如果设置了这些种类中的任何一个，则它们将覆盖较低优先级的 LANG 环境变量。有关环境变量分层结构的详细信息，请参见 setlocale(3C) 手册页。

您也可以从 CDE 桌面上启动 en_US.UTF-8 环境。在 CDE 登录屏幕的 Options -> Language 菜单上，选择 en_US.UTF-8。

TTY 环境设置

根据所使用的终端和终端仿真器的不同，您可能需要将某些特定代码集的 STREAMS 模块 Push 到流上。

有关 STREAMS 模块和一般流的详细信息，请参见《*STREAMS Programming Guide*》。

下表 显示在终端环境中受 en_US.UTF-8 语言环境支持的 STREAMS 模块。

表 5–7 en_US.UTF-8 所支持的 32 位 STREAMS 模块

32 位 STREAMS 模块	说明
/usr/kernel/strmod/u8lat1	UTF-8 和 ISO8859-1 （西欧语言）之间的代码转换 STREAMS 模块
/usr/kernel/strmod/u8lat2	UTF-8 和 ISO8859-2 （东欧语言）之间的代码转换 STREAMS 模块
/usr/kernel/strmod/u8koi8	UTF-8 和 KOI8-R （西里尔文）之间的代码转换 STREAMS 模块

下表列出 en_US.UTF-8 所支持的 64 位 STREAMS 模块。

表 5–8 en_US.UTF-8 所支持的 64 位 STREAMS 模块

64 位 STREAMS 模块	说明
/usr/kernel/strmod/sparcv9/u8lat1	UTF-8 和 ISO8859-1（西欧语言）之间的代码转换 STREAMS 模块
/usr/kernel/strmod/sparcv9/u8lat2	UTF-8 和 ISO8859-2（东欧语言）之间的代码转换 STREAMS 模块
/usr/kernel/strmod/sparcv9/u8koi8	UTF-8 和 KOI8-R（西里尔文）之间的代码转换 STREAMS 模块

将 STREAMS 模块装入内核

要将 STREAMS 模块装入内核，请首先成为超级用户。

要确定您使用的是 64 位 Solaris 还是 32 位 Solaris 系统，请使用 isainfo(1) 公用程序，如下所示：

```
system# isainfo -v
64-bit sparcv9 applications
32-bit sparc applications
```

如果该命令返回此信息，则您所运行的是 64 位 Solaris 系统。如果运行的是 32 位 Solaris 系统，则此公用程序将显示以下内容：

```
system# isainfo -v
32-bit sparc applications
```

使用 modinfo(1M) 来确保您的系统尚未装入 STREAMS 模块：

```
system# modinfo | grep modulename
```

如果已安装了 STREAMS 模块（如 u8lat1）， 则输出将类似于以下内容：

```
system# modinfo | grep u8lat1
89 ff798000  4b13  18   1  u8lat1 (UTF-8 <--> ISO 8859-1 module)
```

如果该模块已安装，您就无须再装入它了。但如果该模块尚未装入，请使用 modload(1M)，如下所示：

```
system# modload /usr/kernel/strmod/u8lat1
```

该命令将 32 位 u8lat1 STREAMS 模块装入到内核中，这样您就可以将其推 (Push) 入流中。如果您所运行的是 64 位 Solaris 产品，请使用 modload(1M)，如下所示：

```
system# modload /usr/kernel/strmod/sparcv9/u8lat1
```

STREAMS 模块已装入到内核中，此时您就可以将其推入到流中了。

要从内核中卸下一个模块，请使用 modunload(1M)，如下所示。在此示例中，u8lat1 模块正被卸下。

```
system# modinfo | grep u8lat1
89 ff798000  4b13  18   1  u8lat1 (UTF-8 <--> ISO 8859-1 module)
system# modunload -i 89
```

能够输入和输出 UTF-8 字符的 dtterm 和终端

与 Solaris 操作环境以前的版本不同，支持输入和输出 UTF-8 代码集的 dtterm(1) 终端和任何其它终端在它们的流中无需任何附加的 STREAMS 模块。ldterm(7M) 模块现在是与代码集无关的，并且还支持 Unicode/UTF-8。

要为 Unicode 语言环境设置正确的终端环境，请使用 stty(1) 公用程序。 要查询当前的设置，请使用 stty(1) 公用程序的 -a 选项，如下所示：

```
system% /bin/stty -a
```

注意：由于 /usr/ucb/stty 未经国际化，因此请改用 /bin/stty。

Latin-1、Latin-2 或 KOI8-R 的终端支持

对于仅支持 Latin-1 (ISO8859-1)、Latin-2 (ISO8859-2) 或 KOI8-R 的终端，您应具有下列 STREAMS 配置：

```
head <-> ttcompat <->  ldterm <->  u8lat1 <-> TTY
```

此配置仅适用于支持 Latin-1 的终端。 对于 Latin-2 终端，请将 STREAMS 模块 u8lat1 替换为 u8lat2。对于 KOI8-R 终端，请将该模块替换为 u8koi8。

确保已将 STREAMS 模块装入到了内核中。

要设置以上显示的 STREAMS 配置，请使用 strchg (1M)，如此示例的第二个命令行所示：

```
system% cat> tmp/mystreams
ttcompat
ldterm
u8lat1
ptem
^D
system% strchg -f /tmp/mystreams
```

确保在使用 strchg(1) 时您是超级用户或此设备的属主。要查看当前配置，请使用 strconf(1)，如下所示：

```
system% strconf
ttcompat
ldterm
u8lat1
ptem
pts
system%
```

要复位原始配置，请按下列方式设置 STREAMS 配置：

```
system% cat> /tmp/orgstreams
ttcompat
ldterm
ptem
^D
system% strchg -f /tmp/orgstreams
```

将设置保存在 ~/.cshrc 中

假定必要的 STREAMS 模块已装入到内核中，您可以将下列各行保存在 .cshrc 文件（C shell 示例）中以便于使用：

```
setenv LANG en_US.UTF-8
if ($?USER != 0 && $?prompt != 0) then
     cat>! /tmp/mystreams$$ << _EOF
     ttcompat
     ldtterm
     u8lat1
     ptem
_EOF
     /bin/strchg -f /tmp/mystreams$$
     /bin/rm -f /tmp/mystreams$$
     /bin/stty cs8 -istrip defeucw
endif
```

在 .cshrc 文件中有了这几行后，您就不必在每次使用 STREAMS 模块时都要键入所有这些命令了。请注意，第二个 _EOF 应从该文件的第一列开始。

代码转换

Unicode 语言环境支持通过 iconv(1)、iconv(3C) 和 sdtconvtool(1) 在多个国家/地区的主要代码集之间增加了各种代码转换。

在 Solaris 9 环境下，公用程序 geniconvtbl 可启用用户定义的代码转换。用 geniconvtbl 公用程序创建的用户定义代码转换可以同时用于 iconv(1) 和 iconv (3)。 有关该公用程序的详细信息，请参考 geniconvtbl(1) 和 geniconvtbl(4) 手册页。

附录 A 中的表中显示了可应用于iconv(1)、iconv_open (3C) 和 sdtconvtool(1) 的可用 fromcode 和 tocode 名称。 有关 iconv 代码转换的详细信息，请参见 iconv(1)、iconv_open(3C)、iconv(3C)、iconv_close(3C)、 geniconvtbl(1) geniconvtbl(4) 和 sdtconvtool(1) 手册页。 有关可用代码转换的详细信息，请参见 iconv_en_US.UTF-8(5)、iconv(5)、iconv_ja (5)、iconv_ko(5)、iconv_zh (5) 和 iconv_zh_TW(5) 手册页。 另见 附录 A。

注意：UCS-2、UCS-4、UTF-16 和 UTF-32 均为 Unicode/ ISO/IEC 10646 表示形式，用于当 Unicode 3.1 和 ISO/IEC 10646-1:2000 标准定义的 Byte Order Mark (BOM) 字符出现在字符流的开头时对这些字符加以识别。其它形式（如 UCS-2BE、UCS-4BE、UTF-16BE 和 UTF-32BE）均为固定宽度的 Unicode/ISO/IEC 10646 表示形式，它们不识别 BOM 字符，并且同样采用大尾数法字节排序。其它表示形式（如 UCS-2LE、UCS-4LE、UTF-16LE 和 UTF-32LE）将采用小尾数法字节排序。它们也不识别 BOM 字符。

有关 ISO8859–* 和 KO18–* 的关联语言 (script) 和语言的信息，请参见 http://czyborra.com/charsets/iso8869.html。

DtMail

由于语言的覆盖范围增加，因此在 `en_US.UTF-8` 语言环境中运行的 Solaris 9 DtMail 支持下列字符集（按 MIME 名称指示）：

- US-ASCII（7 位的 US ASCII）
- UTF-8（UCS 传输格式 8 位）
- UTF-7（ (UCS 传输格式 7 位）
- ISO-8859-1（拉丁语-1）
- ISO-8859-2（拉丁语-2）
- ISO-8859-3（拉丁语-3）
- ISO-8859-4（拉丁语-4）
- ISO-8859-5（拉丁文/西里尔语）
- ISO-8859-6（拉丁文/阿拉伯语）
- ISO-8859-7（拉丁文/希腊语）
- ISO-8859-8（拉丁文/希伯来语）
- ISO-8859-9 (Latin-5)
- ISO-8859-10 (Latin-6)
- ISO-8859-13（Latin-7/波罗的海语）
- ISO-8859-14（Latin-8/凯尔特语）
- ISO-8859-15 (Latin-9)
- ISO-8859-16 (Latin-10)
- KOI8-R（西里尔语）
- ISO-2022-JP 和 EUC-JP（日语）
- ISO-2022-KR 和 EUC-KR（朝鲜语）
- ISO-2022-CN（简体中文）
- ISO-8859–13（Latin-7/波罗的海语）
- ISO-8859–14（Latin-8/凯尔特语）
- KOI8–U（西里尔语/乌克兰语）
- Shift_JIS（Shift JIS 中的日语)
- GB2312（EUC 中的简体中文）
- TIS-620（泰语）
- UTF-16（UCS 传输格式 16 位）
- UTF-16BE（UTF-16 大尾数法）
- UTF-16LE（UTF-16 小尾数法）
- Windows-1250
- Windows-1251
- Windows-1252
- Windows-1253
- Windows-1254
- Windows-1255
- Windows-1256
- Windows-1257
- Windows-1258
- Big5（繁体中文）
- UTF-32（UCS 传输格式 32 位）

- UTF-32BE（UTF-32 大尾数法）
- UTF-32LE（UTF-32 小尾数法）

这种支持支持用户在一个 DtMail 示例中查看以世界上几乎任何地区的不同字符集编码的任何种类的电子邮件。DtMail 通过查看 MIME 字符集以及随同电子邮件提供的内容传送编码来对收到的电子邮件进行解码。支持 Windows-125x MIME 字符集。

要发送电子邮件，您必须指定一个收件人邮件用户代理（邮件客户）能够理解的 MIME 字符集，也可以使用 `en_US.UTF-8` 语言环境提供的缺省 MIME 字符集。可以在"新建邮件"窗口中切换外发邮件的字符集，方法是按下 `Control Y`，或者单击"格式"菜单按钮，然后单击"更改字符集"按钮。下一个可用的字符集名称将显示在"发送"按钮顶部的左下角区域内。

如果您的电子邮件信息标题或信息正文包含无法由指定的 MIME 字符集表示的字符，则系统会自动将字符集切换为可表示任何字符的 `UTF-8`。

如果您的信息仅包含来自 7 位 US-ASCII 字符集的字符，则电子邮件的缺省 MIME 字符集将为 `US-ASCII`。任何邮件用户代理都可以对这种电子邮件信息进行解码，而不会丢失字符或信息。

如果您的信息包含来自各种语言的字符，则缺省 MIME 字符集将为 `UTF-8`。 `UTF-8` 中的任何 8 位字符都是用 Quoted-Printable 编码来编码的。有关 MIME、已注册的 MIME 字符集和 Quoted-Printable 编码的详细信息，请参考 RFC 2045、2046、2047、2048、2049、2279、2152、2237、1922、1557、1555 和 1489。



图形 5–12 DtMail 的“新建邮件”窗口

程序设计环境

国际化的应用程序应自动启用 en_US.UTF-8 语言环境。但在应用程序的资源文件中需要有正确的 FontSet/XmFontList 定义。

有关国际化应用程序的信息，请参见《**创建适用于全世界的软件：** *Solaris* **国际开发者指南**》，第 2 版。

用于 X 应用程序的 FontSet

有关用于 X 应用程序的 FontSet 的信息，请参见 第 92 页“Unicode 语言环境 ：en_US.UTF-8 支持概述”。

由于 Solaris 9 环境支持 CDE 桌面环境，因此每一种字符集都有一组有保证的字体。

以下是在 Solaris 9 产品中受支持的 Latin-1 字体的列表：

```
-dt-interface system-medium-r-normal-xxs sans utf-10-100-72-72-p-59-iso8859-1
-dt-interface system-medium-r-normal-xs sans  utf-12-120-72-72-p-71-iso8859-1
-dt-interface system-medium-r-normal-s sans  utf-14-140-72-72-p-82-iso8859-1
-dt-interface system-medium-r-normal-m sans  utf-17-170-72-72-p-97-iso8859-1
-dt-interface system-medium-r-normal-l sans  utf-18-180-72-72-p-106-iso8859-1
-dt-interface system-medium-r-normal-xl sans utf-20-200-72-72-p-114-iso8859-1
-dt-interface system-medium-r-normal-xxl sans utf-24-240-72-72-p-137-iso8859-1
```

有关 CDE 公共字体别名（包括 -dt-interface user-* 和 -dt-application-* 别名）的信息，请参见《*Common Desktop Environment: Internationalization Programmer's Guide*》。

在 en_US.UTF-8 语言环境中，utf 也作为 X 逻辑字体描述名的样式字段中的附加属性包括在该语言环境的公共字体别名中。因此，要有一组正确的字体，在字体集创建过程中就必须包括附加样式，如下例所示：

```
fs = XCreateFontSet(display,
"-dt-interface system-medium-r-normal-s*utf*",
 &missing_ptr, &missing_count, &def_string);
```

CDE/Motif 应用程序中的 FontList 定义

与 FontSet 定义一样，应用程序的 XmFontList 资源定义也应包括该语言环境所支持的附加样式属性。

```
*fontList:\
 -dt-interface system-medium-r-normal-s*utf*:
```

第 6 章

复杂文本布局

复杂文本布局 (CTL) 扩展使得 Motif API 能够支持某些写入系统，这些写入系统要求在逻辑和物理文本表示之间进行复杂变换，如阿拉伯语、希伯来语和泰语所要求的变换。CTL Motif 提供了字符成形，如连字、音调符号和段排序，并且支持静态和动态文本小部件的变换。它还支持从右到左和从左到右的文本定向以及动态文本小部件的制表。由于文本绘制是通过绘制层处理的，因此可以很容易地扩展其它小部件库以支持 CTL。

CTL 技术概述

要充分利用这些新特征，用户必须有可移植布局服务 (PLS) 库和适当的语言引擎。CTL 使用 PLS 作为语言引擎的接口，并在显示文本前使用语言引擎变换文本。支持 CTL 的应用程序必须包含附加的资源，请见 CTL 文档中的说明。

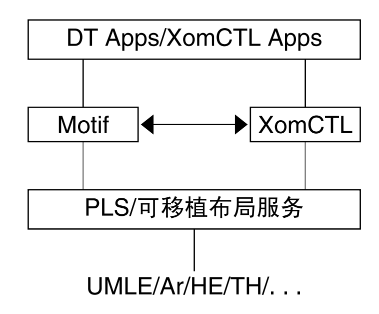
具体说来，XomCTL 支持下列复杂语言成形 和重新整理特征（这些特征是由基础的与语言环境有关的 PLS 模块变换提供的）：

- 位置变化
- 绑扎（多到一）和字符组合（一到多）
- 音调符号
- 双向性
- 对称交换
- 数字成形
- 字符串验证

CTL 体系结构概述

CTL 体系结构 的组织如 图形 6–1 中所示。 堆栈顶部的 `Dt Apps` 使用 Motif CTL 功能来绘制文本。而 Motif 则使用 PLS 与特定语言环境的语言引擎交互，并执行变换以支持位置变化、数字成形等等。

CTL 体系结构被设计成通过添加新的特定于语言环境的引擎来支持新的语言。换句话说，无须改变 Motif 或 `Dt Apps` 即可添加对泰语和越南语的支持。



图形 6–1 CTL 体系结构

对基于 X 库的应用程序的 CTL 支持

XomCTL（X 库输出模块中的复杂文本布局支持）允许所有的纯粹 X Windows 应用程序（如基于 X 的终端仿真器）具有 CTL 支持。XomCTL 提供了包括 X11 哑字体支持在内的具有完全特色的 Open Source XI18N 实现。

新的 XOC 资源

Solaris 9 环境提供了下列 XOC 资源：

`XNText`	使用户能够设置需要在其中执行 CTL 操作的文本缓冲区。
`XNTextLayoutNumGlyphs`	为文本缓冲区中的文本提供字型数目。
`XNTextLayoutModifier`	与 XmNLayoutModifier 或 Motif 相同。
`XNTextLayoutProperty`	与PLS 特性相同（输入至输出和输出至输入）。

`XNTextLayoutMapInpToOut`	与PLS 特性相同（输入至输出和输出至输入）。
`XNTextLayoutMapOutToInp`	与PLS 特性相同（输入至输出和输出至输入）。

这些特征的描述可以从 X/Open 或PLS 可移植布局服务的规范中获得。

支持 CTL 技术的 Motif 更改

下列 Motif 更改支持 CTL 技术。

`XmNlayoutDirection`	控制对象布局。
`XmStringDirection`	指定系统显示字符串中的字符的方向。
`XmRendition`	将新的伪资源添加到 `XmRendition`。
`XmText` 和 `XmTextField`	影响与 `XmRendition`关联的文本的布局行为。
`XmTextFieldGetLayoutModifier`	返回绘制布局对象的布局修饰符字符串。
`XmTextGetLayoutModifier`	返回与小部件关联的绘制的当前布局对象设置值。
`XmTextFieldSetLayoutModifier`	设置与其绘制关联的布局对象的布局修饰符值。
`XmTextSetLayoutModifier`	修改与小部件关联的绘制的布局对象设置。
`XmStringDirectionCreate`	创建复合字符串。

XmNlayoutDirection

`XmNlayoutDirection` 资源[1] 控制对象布局。它以如下所述的方式同 `LayoutObject` 的定向值进行交互。

1 有关 `XmNlayoutDirection` 的概述，尤其是对 `XmStringDirection` 和 `XmNlayoutDirection` 之间交互的说明，请参见 Motif《**程序员指南**》（2.1 版 ）的 11.3 节。

确定布局方向

如果将 XmNlayoutDirection 指定为 XmDEFAULT_DIRECTION，则小部件的布局方向在创建时从主导伪 XOC 进行设置。对于动态文本（XmText 和 XmTextField），主导伪 XOC 是与用于小部件的 XmRendition 关联的伪 XOC。对于静态文本 （XmList、XmLabel、XmLabelG），布局方向从指定方向的第一个复合字符串组件设置。此规范以下列两种方式之一发生：

直接 组件类型为 XmSTRING_COMPONENT_LAYOUT_PUSH 或 XmSTRING_COMPONENT_DIRECTION.

间接 组件类型为 XmSTRING_COMPONENT_LOCALE_TEXT、XmSTRING_COMPONENT_WIDECHAR_TEXT 或 XmSTRING_COMPONENT_TEXT，它们都来自组件的关联 XmRendition 和关联 LayoutObject。

如果未将 XmNlayoutDirection 指定为 XmDEFAULT_DIRECTION，并且未在布局修饰符字符串中显式指定 XmNlayoutModifier @ls orientation 值，则 XmNlayoutDirection 值将传递给 XOC 及其 LayoutObject。

如果同时显式指定了 XmNlayoutDirection 和 XmNlayoutModifier @ls orientation，则行为将混合在一起。XmNlayoutDirection 控制小部件对象布局，而 XmNlayoutModifier @ls orientation 值控制布局变换。

请参见《*CAE* **规范：可移植布局服务：依赖于上下文的定向文本**》。开放组：1997 年 2 月； ISBN 1-85912-142-X；文档号码 C616 — 对用于处理依赖于上下文的双向文本转换的可移植功能的说明，这些功能是对现有 POSIX 语言环境模型的扩展。 该文档适用于希望为复杂文本语言提供支持的系统和应用程序程序员。

XmStringDirection

XmStringDirection 是用于指定系统显示字符串字符的方向的数据类型。

XmNlayoutDirection 资源为任何不包含用于指定字符串方向的组件的复合字符串 (XmString) 设置缺省绘制方向。因此，要设置布局方向，您需要为 XmNlayoutDirection 资源设置适当值。 不需要创建具有特定方向组件的复合字符串。当应用程序绘制 XmString 时，该应用程序应查看所创建的字符串是否具有显式方向 (XmStringDirection)。如果没有方向组件，应用程序应检查当前小部件的 XmNlayoutDirection 资源的值并使用该值作为 XmString 的缺省绘制方向。

XmRendition

CTL 将新的伪资源添加到下表中列出的 XmRendition：

表 6–1 XmRendition 中的新资源

名称c	类/类型	存取	缺省值c
XmNfontType	XmCFontType/XmFontType	CSG	XmAS_IS
XmNlayoutAttrObject	XmClayoutAttrObject/String	CG	NULL
XmNlayoutModifier	XmClayoutModifier/String	CSG	NULL

XmNfontType
指定"绘制"字体对象的类型。对于 CTL，此资源的值必须是 XmFONT_IS_XOC 值。如果该值不匹配，则 XmNlayoutAttrObject 和 XmNlayoutModifier 资源将被忽略。

如果此资源的值是 XmFont_IS_XOC，并且未指定 XmNfont 资源，则 XmNfontName 资源的值将在创建时在 XmNlayoutAttrObject 资源所指定的语言环境或当前语言环境中转换成 XOC 对象。 此外， XmNlayoutModifier 资源的值被传递到任何与该 XOC 关联的布局对象。

XmNlayoutAttrObject
指定布局 AttrObject 参数。 该资源用于创建与该 XOC（它与此 XmRendition 关联）关联的布局对象。有关该字符串的语法和语义，请参考布局服务 m_create_layout() 规范。有关"布局修饰符定向"输出值和 XmNlayoutDirection 小部件资源间交互作用的说明，请参见 XmNfontType 的描述。

XmNlayoutModifier
指定要传递给与该 XmRendition 的 XOC 一起使用的布局对象的布局值。 有关该字符串的语法和语义，请参见《*CAE* **规范**》。

使用 XmRendition{Retrieve,Update} 设置该资源可将该字符串传递给与此绘制的关联 XOL 相关联的布局对象。这是用于动态配置布局服务的机制。如果 Orientation、Context、 TypeOfText、TextShaping 或 ShapeCharset 发生更改，则可能会导致不可预测的行为。

附加布局行为

XmNlayoutModifier 影响与 XmRendition 关联的文本的布局行为。 例如，如果数字的布局缺省处理是 NUMERALS_NOMINAL，则用户可以通过将XmNlayoutModifier 设置为 @ls numerals=nominal:national 或 @ls numerals=:national 而更改为 NUMERALS_NATIONAL。

布局值可分为下列几个组：

- 编码说明 —TypeOfText、 TextShaping、ShapeCharset（以及语言环境代码集）

  TypeOfText 从本质上而言是段排序，因此可以使用不透明块进行阐释。通过绘制对象动态修改这些值通常是没有意义的，并且几乎肯定会导致引起不可预测的行为。
- 布局行为 — Orientation、Context、 ImplicitAlg、Swapping 和 Numerals 。 Orientation 和 Context 不应该动态修改。 您可以安全地修改 ImplicitAlg 、Swapping 和 Numerals。
- 编辑行为 — CheckMode

XmText 和 XmTextField

Xm CTL 扩展了 XmText 和 XmTextField，方法是模仿 Motif 2.0 CSText 小部件，添加一组以可视方式运行的相似的移动和删除操作。标准 Motif 2.1 Text 和 TextField 不区分逻辑和物理顺序：**下一个** 和**向前** 意为 "向右"，而**上一个** 和**向后** 意为"向左" CSText，然而，请进行适当的区分并定义一组具有严格物理名称的新操作（例如，left-character ()、delete-right-word()等等）。所有这些操作例程均被定义为对小部件的 XmNlayoutDirection 敏感，并可调用适当的**下一个** 或**上一个**操作。 Xm CTL 扩展比 CSText 扩展稍微复杂一些。 Xm CTL 扩展对小部件的全局定向不敏感，但是对伪 XOC 所确定的光标周围物理字符的特定方向性（包括中性稳定性）敏感。

还有一个新的资源，用于控制选择策略、提供绘制标签以及控制对齐方式。

新 Xm CTL 操作集合大致是 {Move,Delete,Kill} 乘以 {Left,Right} 乘以 {Character,Word} 的向量积，如下所示。

表 6–2 Xm CTL 中的新资源

名称	类/类型	存取	缺省值
XmNrenditionTag	XmCRenditionTag/XmRString	CSG	XmFONTLIST_DEFAULT_TAG
XmNalignment	XmCAlignment/XmRAlignment	CSG	XmALIGNMENT_BEGINNING
XmNeditPolicy	XmCEditPolicy/XmREditPolicy	CSG	XmEDIT_LOGICAL

XmNrenditionTag
: 指定位于 XmNrenderTable 资源中（用于此小部件）的XmRendition 的绘制标签。

XmNalignment
: 指定此小部件中使用的文本对齐方式。 仅支持 XmALIGNMENT_END 和 XmALIGNMENT_CENTER。

XmNeditPolicy

指定此小部件使用的编辑策略（XmEDIT_LOGICAL 或 XmEDIT_VISUAL）。 如果指定 XmEDIT_VISUAL，则选择、光标移动和删除将以可视形式进行。 设置该资源还可将标准键盘移动和删除事件的转换更改为新的"可视"操作列表或现有的逻辑操作。

字符定向操作例程

forward-cell() 和 backward-cell() 操作在指定的方向中查询字符的定向。 如果方向是从左向右，则它们调用相应的**下一个/向前** 或**上一个/向后** 变量：

字符定向附加行为

这些操作通过使用布局服务变换 OutToInp 和 Property 缓冲区（对于嵌套级）来确定字符的定向。 因而，小部件的行为取决于语言环境的特定变换。 如果 OutToInp （尤其是 Property 缓冲区）中的信息不准确，则小部件可能会产生意外行为。 此外，由于语言环境特定的模块处于该规范的范围之外，相同文本、应用程序、资源值和 LayoutObject 配置的双向编辑行为可能因平台的不同而不同。

可视模式操作导致显示基于单元的行为。逻辑模式操作导致基于字符的逻辑行为。 例如，delete-right-character() 操作删除与显示单元相对应的输入缓冲区字符。 即，一个 LayoutObject 变换 "特性"字节"新单元指示器"为 1 的输入缓冲区字符，以及"新单元指示器"[2] 为 0 的所有后继字符。

同样，对于 backward-character()，插入点在输入缓冲区中向后移动一个字符，光标被在与关联的输出缓冲区字符相对应的可视位置重新绘制。 这意味着需要进行若干次击键以在复合显示单元中移动；当插入点在"新单元指示器"是 0 的输入缓冲区字符（即音调符号或连字片段）中移动时，光标并不实际更改显示位置。

这意味着删除操作或者从逻辑/输入缓冲区端执行，或者从物理/输出端的显示单元级执行。 没有严格的、物理的逐字符删除模式，因为在输入和输出缓冲区之间不存在一对一的对应关系。 例如，给定的物理字符只能代表逻辑字符片断。

XmText 操作例程

下表描述 XmText 操作例程。

left-character(extend)

如果 XmNeditPolicy 为 XmEDIT_LOGICAL 并且被不带参数调用，则插入光标在逻辑上向后移动一个字符。 如果插入光标位于行首，则它将移动到上一行（如果存在的话）的最后一个逻辑字符。 否则，插入光标位置不变。

如果 XmNeditPolicy 为 XmEDIT_VISUAL，则光标移动到光标位置的左侧。 如果插入光标位于行首，则它将移动到上一行（如果存在的话）的结尾字符。

2 有关 Property 缓冲区的详细信息，请参见《*CAE* **规范** 》中的 m_transform_layout() 规范。

如果使用 extend 参数调用 left-character()，则插入光标就象没有参数时一样移动，并扩展当前所选内容。

left-character() 操作使用原因值 XmCR_MOVING_INSERT_CURSOR 生成对 XmNmotionVerifyCallback 过程的调用。 如果使用 extend 参数调用，则该操作可以生成对 XmNgainPrimaryCallback 过程的调用。 有关详细信息，请参见《*Motif* **程序员参考**》中的回调说明。

right-character(extend)
如果 XmNeditPolicy 为 XmEDIT_LOGICAL 并且被不带参数调用，则它将插入光标逻辑地向前移动一个字符。 如果插入光标位于行的逻辑结尾处，则它将插入光标移动到下一行（如果存在的话）的逻辑开始处。

如果 XmNeditPolicy 为 XmEDIT_VISUAL，则光标移动到该光标位置的右侧。 如果插入光标位于行尾，则它将插入光标移动到下一行（如果存在的话）的开始处。

如果使用 extend 参数调用，则它将象没有参数一样移动插入光标并扩展当前所选内容。

right-character() 操作使用原因值 XmCR_MOVING_INSERT_CURSOR 生成对 XmNmotionVerifyCallback 过程的调用。 如果使用 extend 参数进行调用，则可生成对 XmNgainPrimaryCallback 过程的调用。 有关详细信息，请参见 *Motif* **程序员参考** 中的回调说明。

right-word(extend)
如果 XmNeditPolicy 为 XmEDIT_LOGICAL 并且被不带任何参数调用，则它将插入光标移动到逻辑后继字（如果存在的话）的逻辑开始处；否则，它移动到当前字的逻辑结尾处。 如果插入光标位于行的逻辑结尾处或该行的最后一个逻辑字时，它将光标移动到下一行（如果存在的话）的第一个逻辑字；否则，它移动到当前字的逻辑结尾处。

如果 XmNeditPolicy 为 XmEDIT_VISUAL 并被不带参数调用，则它将插入光标移动到行尾右侧或后面的第一个空格字符后的第一个非空格字符处。

如果使用 extend 参数进行调用，则它将象不带参数调用一样移动插入光标，并扩展当前所选内容。

left-word() 操作使用原因值 XmCR_MOVING_INSERT_CURSOR 生成对 XmNmotionVerifyCallback 过程的调用。 如果使用 extend 参数调用，则可以生成对 XmNgainPrimaryCallback 过程的调用。 有关详细信息，请参阅 *Motif* **程序员参考** 中的回调说明。

delete-left-character()
如果 XmNeditPolicy 为 XmEDIT_LOGICAL，它等效于 delete-previous-char ()。 如果 XmNeditPolicy 为 XmEDIT_VISUAL，则在标准模式下，如果有一个非空选择，则它将删除该选择；否则它将删除插入光标左侧的字符。 在添加模式下，如果有一个非空选择，光标未离开所选内容，并且 XmNpendingDelete 被设置为真，则它将删除该选择；否则，它将删除插入光标左侧的字符。 这可能会影响选择。

delete-left-character() 操作使用原因值 XmCR_MODIFYING_TEXT_VALUE 生成对 XmNmodifyVerifyCallback 过程的调用，并且使用原因值 XmCR_VALUE_CHANGED 生成对 XmNvalueChangedCallback 过程的调用。

delete-right-character()
如果 XmNeditPolicy 为 XmEDIT_VISUAL，则它等效于 delete-next-character()。如果 XmNeditPolicy 为 XmEDIT_VISUAL，则在标准模式下，如果有一个非空选择，则它将删除该选择；否则，它删除插入光标右侧的字符。 在添加模式中，如果有一个非空选择，并且光标未离开所选内容，则将 XmNpendingDelete 设置为真，并删除该选择；否则，将删除插入光标右侧的字符。 这可能会对选项造成影响。

delete-right-character() 操作使用原因值 XmCR_MODIFYING_TEXT_VALUE 生成对 XmNmodifyVerify-Callback 过程的调用，并且使用原因值 XmCR_VALUE_CHANGED 生成对XmNvalue-ChangedCallback 过程的调用。

实现了一些基于单元的例程以支持字符组合、连字和音调符号。 换句话说，两个或多个字符可以被占用一个表示单元的单个字形表示。

XmText 单元操作例程如下所示：

backward-cell(extend)
将插入光标向后移动一个单元。 如果 XmNeditPolicy 为 XmEDIT_LOGICAL，则将插入光标移动到逻辑上位于当前单元之前的那个单元（如果存在的话）的开始处。否则，将其移动到当前单元的开始处。

如果 XmNeditPolicy 是 XmEDIT_VISUAL，则光标移动到光标左侧单元（如果它存在的话）的开始处。 prev-cell() 操作使用原因值 XmCR_MOVING_INSERT_CURSOR 生成对 XmNmotionVerifyCallback 过程的调用。 如果使用 extend 参数进行调用，则可以生成对 XmNgainPrimaryCallback 过程的调用。 有关详细信息，请参阅 *Motif* **程序员参考** 中的回调说明。

forward-cell(extend)
将插入光标移动到下一个逻辑单元（如果存在的话）的开始处。否则将它移动到该单元的结尾处。 如果 XmNeditPolicy 是 XmEDIT_LOGICAL，则光标向前移动一个单元。

如果 XmNeditPolicy 是 XmEDIT_VISUAL，则光标移动到该光标位置右侧单元的开始处（如果该单元存在的话）。否则，光标移动到当前单元的结尾处。forward-cell() 操作使用原因值 XmCR_MOVING_INSERT_CURSOR 生成对 XmNmotionVerifyCallback 过程的调用。 如果使用 extend 参数进行调用，则可以生成对 XmNgainPrimaryCallback 过程的调用。 有关详细信息，请参阅 *Motif* **程序员参考** 中的回调说明。

XmTextFieldGetLayoutModifier

XmTextFieldGetLayoutModifier() 返回一个布局修饰符字符串，它反映了与其绘制关联的布局对象的状态。

XmTextFieldGetLayoutModifier() 的语法是：

```
#include <Xm/TextF.h>
String XmTextFieldGetLayoutModifier( 小部件 小部件)
```

XmTextFieldGetLayoutModifier() 存取与小部件关联的绘制的当前布局对象设置值。当使用公用函数更改布局对象修饰符的值时，XmTextFieldGetLayoutModifier 函数返回布局对象的完整状态，而不仅仅返回更改后的值。

XmTextFieldGetLayoutModifier() 以字符串值的形式返回布局对象修饰符值。

XmTextGetLayoutModifier

XmTextGetLayoutModifier() 返回一个布局修饰符字符串，它反映了与其绘制关联的布局对象的状态。

XmTextGetLayoutModifier() 的语法是：

```
#include <Xm/Text.h>
String XmTextGetLayoutModifier( 小部件 小部件 )
```

XmTextGetLayoutModifier 存取与小部件关联的绘制的当前布局对象设置值。当使用公用函数更改布局对象修饰符值时， XmTextGetLayoutModifier 函数返回布局对象的完整状态，而不仅仅返回更改后的值。

XmTextGetLayoutModifier 以字符串值的形式返回布局对象修饰符值。

XmTextFieldSetLayoutModifier

XmTextFieldSetLayoutModifier() 设置布局修饰符值，这会更改与其绘制关联的布局对象的行为。

XmTextFieldSetLayoutModifier() 的语法是：

```
#include <Xm/TextF.h>
void XmTextFieldSetLayoutModifier( 小部件 小部件 ,string
layout_modifier)
```

XmTextFieldSetLayoutModifier 修改与小部件相关联的绘制的布局对象设置。 当使用该公用函数设置布局对象修饰符值时，只有在输入参数中指定的属性被更改；其余属性保持不变。

XmTextSetLayoutModifier

XmTextSetLayoutModifier() 设置布局修饰符值，这会更改与其绘制关联的布局对象的行为。

XmTextSetLayoutModifier() 的语法是：

```
#include <Xm/Text.h>
void XmTextSetLayoutModifier( 小部件 小部件 ,string layout_modifier)
```

XmTextSetLayoutModifier 修改与小部件关联的绘制的布局对象设置。 当布局对象修饰符值使用该公共函数设置时，只有输入参数中指定的属性被更改； 其余属性保持不变。

XmStringDirectionCreate

XmStringDirectionCreate 用于创建复合字符串。

XmTextSetLayoutModifier() 的语法是：

```
#include <Xm/Xm.h>
XmString XmStringDirectionCreate( 方向)
XmStringDirection 方向
```

XmStringDirectionCreate 创建一个具有单个组件（一个具有给定值的方向）的复合字符串。 另一方面，XmNlayoutDirection 资源为任何不具有指定其方向的组件的复合字符串 (XmString) 设置缺省绘制方向。 因此，要设置布局方向，您需要做的全部事情就是为 XmNlayoutDirection 资源设置适当的值。 您不需要创建具有特定方向组件的复合字符串。当应用程序绘制 XmString 时，它应查看所创建的字符串是否具有显式方向 (XmStringDirection)。 如果没有方向组件，应用程序应该检查当前小部件的 XmNlayoutDirection 资源的值并使用该值作为 XmString 的缺省绘制方向。

UIL Arguments

下表显示了 UIL 参数。

表 6–3 UIL

UIL 参数名称	参数类型
XmNlayoutAttrObject	字符串
XmNlayoutModifier	字符串
XmNrenditionTag	字符串
XmNalignment	整数
XmNeditPolicy	整数

如何开发 CTL 应用程序

以下部分说明如何开发 CTL 应用程序。

布局方向

复合字符串的方向被存储起来，以便使数据结构对于描述从左到右的语言（如英语、西班牙语、法语和德语）的文本或者从右到左的语言（如希伯来语和阿拉伯语）的文本同样有用。在 Motif 应用程序中，可以使用 VendorShell 或 MenuShell 中的 XmNlayoutDirection 资源来设置布局方向。管理器和原始小部件（以及小配件）也具有 XmNlayoutDirection 资源。缺省值从具有相同资源的最近的祖先继承。

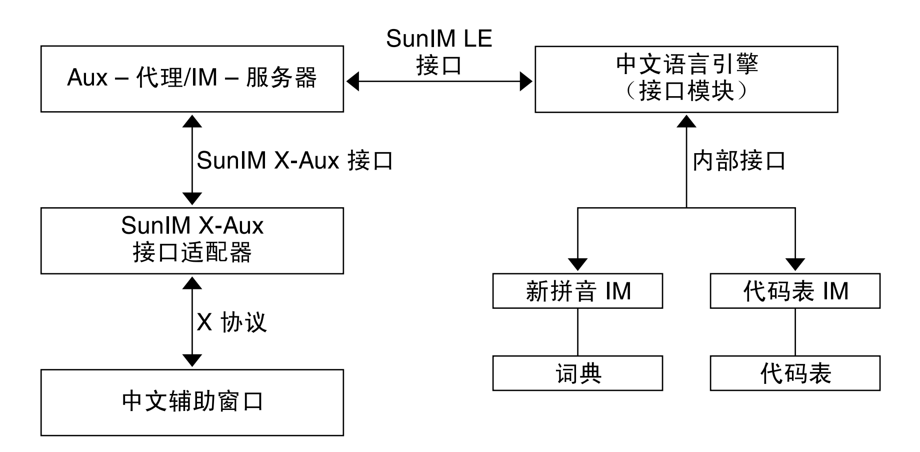
对于 XmText 小部件，还必须指定垂直方向。将 layoutDirection 设置为 XmRIGHT_TO_LEFT 会使字符串方向成为从右到左，而光标则垂直向下移动。如果垂直方向很重要，并且需要从上到下对齐，那么请一定指定 XmRIGHT_TO_LEFT_TOP_TO_BOTTOM ，它指定组件先从右到左排列，再从上到下排列，从而得到所需要的行为。

此外，XmText 和 TextField小部件的行为受 XmRendition的 XmNalignment 和 XmNlayoutModifier 资源的影响。这些资源与 XmNlayoutDirection 共同控制文本小部件的布局行为。图形 6–2 中阐释了此行为。

该图中使用的输入字符串是：

A B ض و

下图的 XmNlayoutModifier 字符串 @ls orientation= 设置值如左列中所示。



图形 6–2 布局方向

如插图所示，XmNAlignment 规定文本是否与布局方向一起右对齐或左对齐。XmNlayoutModifier 将文本分成多段，并根据定向值将它们从左到右或从右到左排列。换句话说，如果 XmNlayoutDirection 是 XmRIGHT_TO_LEFT ，并且 XmNAlignment 值是 XmALIGNMENT_BEGINNING ，则该字符串右对齐。

创建绘制

下面的代码使用 Rendition（其标记为"ArabicShaped"）来创建 XmLabel（其 XmNlabelString 属于 XmCHARSET_TEXT 类型）。Rendition 是用 "ar" （对应于阿拉伯语语言环境的语言环境名称）的 XmNlayoutAttrObject 和布局修饰符字符串（为输出缓冲区指定 NUMERALS_CONTEXTUAL 的 Numerals 值和 "unicode-3.0." 的 ShapeCharset 值）创建的

语言环境的特定布局模块将其输入文本转移到使用 16 位 Unicode 3.0 代码集编码的物理字符的输出缓冲区 。由于已经指定了显式布局语言环境，因此此文本能够独立于运行时语言环境设置正确绘制。 在此示例中，用 ISO 8859–6 对输入进行编码。

```
int n;
Arg args[10];
```

```
Widget w;
XmString labelString;
XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;
      /* alef lam baa noon taa - iso8859-6 */
labelString = XmStringGenerate("\307\344\310\346\312\", NULL
                                     XmCHARSET_TEXT, "ArabicShaped");
w = XtVaCreateManagedWidget("a label", xmLabelWidgetClass, parent,
                             XmNlabelString, labelString,
                                  XmNlabelType, XmSTRING,
                             NULL);
n = 0;
XtSetArg(args[n], XmNfontName, "-*-*-medium-r-normal-*-24-*-*-*-*-*-*");
      n++;
XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;
XtSetArg(args[n], XmNlayoutAttrObject, "ar"); n++;
XtSetArg(args[n], XmNlayoutModifier,
          "@ls numerals=:contextual, shapecharset=iso8859-6"); n++;
renditionTag = (XmStringTag) "ArabicShaped";
rendition = XmRenditionCreate(w, renditionTag, argcs
s, n);
renderTable =
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE_MERGE);
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
```

编辑绘制

下面的代码用单个 Rendition 创建 TextField 小部件和 RenderTable。XmNlayoutAttrObject 和 XmNlayoutModifier 伪资源均未指定，因此缺省为 NULL。 该值意味着与 Rendition 关联的布局对象属于缺省语言环境（如果有的话）。

要使此示例正常工作，该语言环境必须设置为其代码集是 ISO 8859-6 并且特定于该语言环境的布局模块可支持 IMPLICIT_BASIC 算法的语言环境。Rendition 的 LayoutObject 的 ImplicitAlg 值已通过 Rendition 的 XmNlayoutModifier 伪资源进行了修改。

```
int n;
Arg args[10];
Widget w;
    XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;
w = XmCreateTextField(parent, "text field", args, 0);
n = 0;
    XtSetArg(args[n], XmNfontName, "-*-*-medium-r-normal-*-24-*-*-*-*-*-*");
     n++;
    XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;
renditionTag = (XmStringTag) "ArabicShaped";
rendition = XmRenditionCreate(w, renditionTag, args, n);
renderTable =
```

```
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE_MERGE);
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
    ....
n = 0;
XtSetArg(args[n], XmNlayoutModifier, "@ls implicitalg=basic");
     n++;
XmRenditionUpdate(rendition, args, n);
```

在资源文件中创建绘制表

对于正确国际化的应用程序，应在其资源文件中指定 Rendition 和绘制表。在文件中指定了绘制表后，程序二进制代码就与给定语言环境的特定需要无关，从而能够轻松地进行定制以满足本地需要。

绘制表是用以下语法在资源文件中指定的： *resource_spec*:[*tag*[, *tag*]*]

其中 *tag* 表示适用于某个绘制的 XmNtag 资源的某个字符串。

该行将按指定要求创建包含一个或多个绘制的绘制表。这些绘制被附加到指定的标记上：

resource_spec[*|.] *rendition*[*|.]*resource_name*:**值**

以下示例阐释了与 XmRendition 相关的 CTL 资源，这些资源可用资源文件来设置。fontType 必须设置为 FONT_IS_XOC 才能使布局对象生效。使用 @ls 指定的 layoutModifier 通过绘制对象传递给布局对象。

有关可使用 layoutModifier 在布局对象上设置的完整资源列表，请参见《*CAE* **规范：可移植布局服务：与上下文有关的定向文本**》，开放组：1997 年 2 月；ISBN 1-85912-142-X；文档号 C616。

在应用程序中创建绘制表

在创建绘制表之前，应用程序必须至少先创建一个属于该表的绘制。顾名思义，XmRenderTableAddRenditions() 函数也用于使用新的绘制补充绘制表。要创建新的绘制表，请使用 NULL 参数调用 XmRenderTableAddRenditions() 函数来替换现有的绘制表。

以下代码将使用在将 XmNfontType 设置为 XmFONT_IS_XOC 时创建的绘制来创建一个绘制表。

```
int n;
Arg args[10];
Widget w;
```

```
XmString labelString;
XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;
      /* alef lam baa noon taa - iso8859-6 */
labelString = XmStringGenerate("\307\344\310\346\312\", NULL
                                       XmCHARSET_TEXT, "ArabicShaped");
w = XtVaCreateManagedWidget("a label", xmLabelWidgetClass, parent,
                            XmNlabelString, labelString,
                                 XmNlabelType, XmSTRING,
                            NULL);
n = 0;
XtSetArg(args[n], XmNfontName, "-*-*-medium-r-normal-*-24-*-*-*-*-*-*-*");
     n++;
XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;
XtSetArg(args[n], XmNlayoutAttrObject, "ar"); n++;
XtSetArg(args[n], XmNlayoutModifier,
          "@ls numerals=nominal:contextual, shapecharset=iso8859-6"); n++;
renditionTag = (XmStringTag) "ArabicShaped";
rendition = XmRenditionCreate(w, renditionTag, args, n);
renderTable =
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE);
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
```

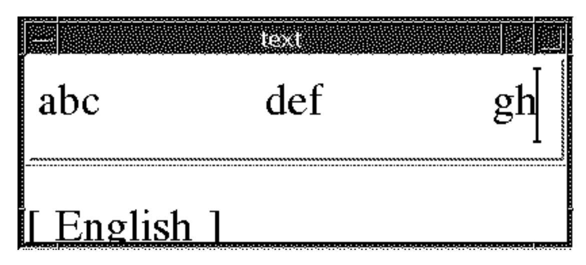
水平制表符

为控制文本的位置，复合字符串可包含几个制表符字符。为了在显示时对这些字符进行解码，一个小部件将指向实际用于该复合字符串的绘制，并找到一个制表停止位置列表。但是，动态小部件 `TextField` 和 `XmText` 并不使用该绘制的制表符资源。它们使用 `8*(width of character 0)` 公式来计算制表符宽度。

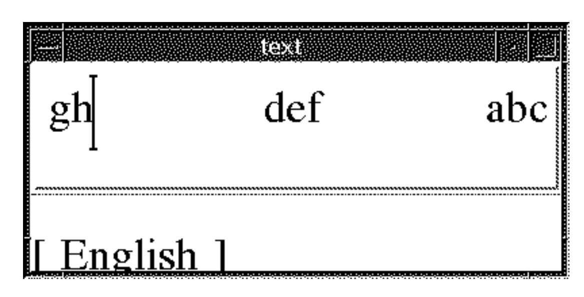
制表符度量值是与复合字符串显示的左边界的距离。如果布局方向是从右向左的，则该距离从右边界开始度量。无论文本方向如何（阿拉伯语从右向左或英语从左向右），制表符都按布局方向 (`XmNlayoutDirection`) 的指定要求向右或向左插入空格。

制表符后面的文本总是在制表停止位置处对齐。制表停止位置从小部件的起始处算起，而小部件的起始处又受到 `XmNlayoutDirection` 的影响。 下图阐释了制表符的行为及其与文本的方向性和小部件的 `XmNlayoutDirection` 的交互作用。

此说明的输入是 `abc\tdef\tgh`。



布局方向：XmLEFT_TO_RIGHT



布局方向：XmRIGHT_TO_LEFT

图形 6–3 制表符行为

鼠标选定

用户用鼠标按钮 1 创建一个主选定。 按下此按钮将撤消选定全部现有的选定，并将插入光标和锚点移到文本中按下按钮的位置上。按住鼠标按钮 1 并拖动它，可以选择锚点和指针位置之间的所有文本，并撤消选定此范围外的任何文本。

选定的文本受资源 XmNeditPolicy 影响，而后者可设置为 XmEDIT_LOGICAL 或 XmEDIT_VISUAL。如果 XmNeditPolicy 设置为 XmEDIT_LOGICAL，并且选定的文本是双向的，则选定的文本从视觉上将是不连续的，而是多个文本段的集合。这是因为逻辑缓冲区中的文本与显示内容之间没有一对一的对应关系。

因此，双向文本的逻辑字符的连续缓冲区在绘制时不能产生连续的字符流。与此相反，如果 XmNeditPolicy 设置为 XmEDIT_VISUAL，则选定文本在视觉上可能是连续的，而在逻辑缓冲区中却是分段的。因此，在同一个光标点处选定、删除和插入双向文本的顺序操作不会产生同一个字符串。

键盘选定

用鼠标进行的选定操作也可以使用键盘来完成。可以使用 Shift-arrow 键的组合来选定文本。

选定的文本受资源 XmNeditPolicy 影响，而后者可设置为 XmEDIT_LOGICAL 或 XmEDIT_VISUAL。 如果 XmNeditPolicy 设置为 XmEDIT_LOGICAL，并且选定的文本是双向的，则选定的文本在视觉上将是不连续的。由于逻辑缓冲区中的文本与显示内容之间没有一对一的关系，因此双向文本中的逻辑字符的连续缓冲区在绘制时不能产生连续的字符流。

与此相反，如果 XmNeditPolicy 设置为 XmEDIT_VISUAL，则选定的文本在视觉上可能是连续的，而在逻辑缓冲区中却是分段的。因此，在同一个光标点处按选定、删除和插入双向文本的顺序操作不会产生同一个字符串。

文本资源和几何图形

与几何图形相关的文本资源如下所示：

- 绘制表 XmNrenderTable，小部件用它来选择显示文本时使用的字体或字体集及其它属性。

  Text 和 Textfield 小部件仅可使用与字体相关的绘制资源，如 XmNfontType 。 这些小部件也可以指定布局对象的属性，如 XmNlayoutAttrObject（通常为语言环境标识符）和 XmNlayoutModifier（用来指定要传递到与此 XmRendition 关联的 XOC 相关的布局对象的布局值）。
- 一种资源 (XmNwordWrap)，该资源指定在文本比小部件宽时是否在字边界处分行。

  在字边界处分行不会将一个新行插入到文本中。如果使用的是草体语言（如阿拉伯语），则当字长度大于小部件长度时，字将折向下一行，但下一行的第一个字符的形状与逻辑缓冲区中的上一个字符无关。

移植说明

为复杂文本布局 (CTL) 启用的新 Motif 库位于 /usr/dt/lib/libXm.so.4 中。 如果您的应用程序链接到 libXm.so.3，那么它不支持 CTL。 ldd app_name 显示应用程序链接到的库。 要移植现有的应用程序以启用 CTL，需要执行以下步骤。

1. 将 -DSUN_CTL 添加到您的 Makefile 中。 该标志很重要，它包括支持 CTL 所需的数据结构。这应在编译期间进行设置。

2. 重新编译现有的应用程序。这一重新编译将自动与 CTL 启用的 Motif 库 `libXm.so.4` 进行链接
3. 将 `XmText.translations` 资源添加到应用程序资源文件中。如果没有这些资源，语言环境的布局引擎将不能启动。
4. 请参考附加在您的文档中的应用程序样例。

注意：在 `fontName` 资源中使用您的语言环境中提供的并适用于您的语言环境的字体名称。

例如，如果您希望在 `XmTextField` 或 `XmText` 小部件中进行基于单元的字符移动（泰语），请按以下方式设置相应小部件的平移：

```
XmText.translations: #override \n\

<Key>osfRight:forward-cell() \n\

<Key>osfLeft:backward-cell() \n\

<Key>osfDelete:delete-next-cell() \n\

<Key>osfBackSpace:delete-previous-cell() \n\
```

第 7 章

具有 mp 的打印过滤器增强模式

本章描述具有 mp 的打印增强模式。 它包括下列主题：

- 第 141 页 “用于 UTF-8 的打印”
- 第 142 页 “mp 打印过滤器增强模式概述”
- 第 143 页 “配置文件的本地化”
- 第 149 页 “依赖于语言环境的 prolog 文件”
- 第 149 页 “定制现有的 prolog 文件并添加新的 prolog 文件”
- 第 149 页 “PostScript 文件定制”
- 第 152 页 “.xpr 文件定制”
- 第 155 页 “创建新的 .xpr 文件”

用于 UTF-8 的打印

Solaris 9 环境中提供了增强的 mp(1) 打印过滤器，可用来打印各种输入文件格式，包括用 UTF-8 编写的纯文本文件。 它使用 Solaris 系统提供的 TrueType 和 Type 1 可缩放字体和 X11 位图字体。它还可以使用打印机常驻字体并且可用作打印服务器客户程序。

该公用程序的输出为标准的 PostScript ， 可以发送给任何 PostScript 打印机。当用作 X Print 服务器客户程序时，mp(1) 还可以输出任何页面描述语言，mp 受打印服务器支持。

要使用该公用程序，请键入下面内容：

```
system% mp 文件名 | lp
```

由于该公用程序接受 stdin 流，因此您还可以将其作为过滤器来使用。

```
system% cat 文件名 | mp | lp
:
```

可将该公用程序设置为一个行式打印机的打印过滤器。例如，下面的命令序列通知打印机服务 LP，打印机 lp1 只接受 mp 格式文件。 此命令还将打印机 lp1 安装在端口 /dev/ttya 上。 有关详细信息，请参阅 lpadmin (1M) 手册页。

```
system# lpadmin -p lp1 -v /dev/ttya -I MP
system# accept lp1
system# enable lp1
```

使用 lpfilter(1M)，可以为过滤器添加该公用程序，如下所示：

```
system# lpfilter -f lp1 -F 路径名
```

该命令通知 LP，可通过名为 **路径名** 的过滤器描述文件使用转换器（此处为 mp）。 **路径名** 包含下面内容：

```
Input types: simple
Output types: MP
Command: /usr/bin/mp
```

使用 /usr/bin/mp 过滤器将缺省类型文件的输入转换为 PostScript 输出。

若要打印 UTF-8 文本文件，请使用下面命令：

```
system% lp -T MP UTF-8-file
```

有关更详细的信息，请参阅 mp 手册页。

mp 打印过滤器增强模式概述

mp 打印过滤器在 Solaris 9 发行版中得到了增强。 最新版的 mp 可以用三种不同的模式进行内部处理，以便在一种语言环境中产生输出文件来打印国际文本。可用的模式包括：

- 处理语言环境特定的字体配置文件 mp.conf 的mp。
- 处理语言环境特定的 PostScript prolog 文件 prolog.ps 的 mp。
- 用作 Xprt（X 打印服务器）客户程序的 mp。

下面部分将描述何时使用特定的打印方法以及 mp 针对这些打印方法所使用的配置和支持文件。

与语言环境特定的字体配置文件 mp.conf 一起使用 mp

如果命令行中没给出 -D 或 -P 选项，则此打印方法为缺省方法，除非 /usr/openwin/lib/locale/$LANG/print 或 /usr/lib/lp/locale/$LANG/mp 目录中存在 prolog.ps 文件。prolog.ps 文件强迫 mp 使用文件中的 PostScript 嵌入字体进行打印。即使语言环境中存在 prolog.ps，使用 -M 选项也会忽略 prolog.ps 文件而使用 mp.conf 文件（如果它存在的话）。

此方法使用 /usr/lib/lp/locale/$LANG/mp/mp.conf 字体配置文件。您可能不必更改此文件，除非您需要使用其它替换字体进行打印。此文件可使用 TrueType、Type 1 或pcf 字体进行配置。/usr/lib/lp/locale/C/ 包含此打印模式和下一个方法的公用打印页布局文件 .ps。第 149 页 "定制现有的 prolog 文件并添加新的 prolog 文件" 中提供了如何定制这些文件的说明。

与语言环境特定的 PostScript 描述文件一起使用 mp

如果命令行中没给出 -D 或 -P 选项，并且存在 /usr/openwin/lib/locale/$LANG/print/prolog.ps ，则 prolog.ps 文件将被前置到输出中。根据 .ps prolog 页的打印样式，布局文件也被前置到输出中。

此打印方法仅使用 PostScript 字体文件。第 143 页 "将 mp 用作 Xprt（X 打印服务器）客户程序" 中描述了 prolog.ps 文件的定制。

将 mp 用作 Xprt（X 打印服务器）客户程序

此项支持使 mp 能够为连接到网络（受 X 打印服务器支持）的任何打印机进行打印输出。PostScript 和许多 PCL 版本也由此命令支持。

如果使用了 -D 或 -P 命令选项，并且您的环境中没有设置任何 XPDISPLAY 变量，则打印服务器启动脚本将在运行客户程序的计算机端口 2100 处启动 Xprt 服务器。该脚本还在 mp 完成后终止打印服务器。如果设置了 XPDISPLAY，mp 客户程序将尝试同运行于 XPDISPLAY 的打印服务器进行联系。 在这种情况下，如果服务器没有运行，则不会对它进行启动尝试。

/usr/lib/lp/locale/C/mp 目录包含用作 Xprt 客户程序的mp 的 .xpr 打印页布局文件。 这些是为 300 dpi 打印机创建的样本文件。如果目标打印机具有不同的 dpi 值，该 dpi 值将自动转换为目标打印机的分辨率。

配置文件的本地化

配置文件为添加或更改字体项或字体组项提供灵活性。

系统缺省配置文件 /usr/lib/lp/locale/$LANG/mp/mp.conf，其中 $LANG 是发生打印的语言环境中的语言环境变量。用户可具有一个个人配置文件，该配置文件由 -u *config.file* **路径** 选项指定。

为了便于兼容，连字或变体字型已被编码为字符，这种形式称为 **变形显现形式**。mp.conf 文件主要用于将一种语言环境中的中间代码点映射为用来打印该代码点的字体编码中的变形显现形式。

中间代码点可以是宽字符，也可以是可移植布局服务 (PLS) 层的输出。 复杂文本布局打印要求中间代码点为 PLS 输出。由 mp(1) 生成的缺省中间代码为 PLS 输出。

当前受支持的字体格式是可移植已编译格式 (PCF)、TrueType 和 Type1 格式。既支持系统常驻 Type1 字体又支持打印机常驻 Type1 字体。请记住下列有关 mp.conf 配置文件的格式和内容：

- 行必须以有效关键字（指令）开头。
- 关键字的参数必须与关键字出现在同一行。
- 以 # 字符开头的行被认为是注释，注释延续至该行末尾。
- 以 0x 开头的数字参数被认为是十六进制数。

mp.conf 文件中的不同部分包括：

- 字体别名
- 字体组定义
- 从中间代码范围映射到语言环境中的字体组
- 将每种字体与共享对象相关联，此共享对象把中间代码点映射到字体编码中的变形显现形式

字体别名

mp.conf 文件的字体别名部分用于定义打印时使用的每种字体的别名。这部分的每一行都具有以下形式：

FontNameAlias **字体别名 字体名称 字体路径**

字体别名

为字体名称起别名的常用惯例是，指定字体的编码/脚本名称，名称后面跟一个字母，表示该字体是 Roman 字体、粗体、斜体或粗斜体（R、B、I 或 BI）。

例如，由于 /usr/openwin/lib/X11/fonts/75dpi/courR18.pcf.Z，是 iso88591 Roman 字体，因此给它起的别名为 iso88591R。

字体类型

为 .pcf 字体指定 PCF 、为 Adobe Type1 字体指定 Type1 并为 truetype 字体指定 TrueType 。 在此 mp.config 文件中只能配置这三种字体。

字体路径

请在此处给出字体文件的绝对路径名。对于 Type1 打印机常驻字体，只指定字体名称，如 Helvetica。

例如，

```
FontNameAlias    prnHelveticaR    Type1    Helvetica
```

字体组定义

您可以将同一类型的字体合并为一个字体组。字体组的格式如下：

关键字 `FontGroup`。

字体组名称 字体的组名称。

组类型 字体类型。只为同一类型的字体（PCF、Type1 和 TrueType）创建字体组。

Roman 字体组中的 Roman 字体名称。

粗体 字体组中的粗体字体名称。

斜体 字体组中的斜体字体名称。

粗斜体 字体组中的粗斜体字体名称。

对于创建组，只需要 Roman 字体项。粗体、斜体和粗斜体字体均为可选。邮件／新闻文章的标题可用不同类型的字体来显示，例如。如果只定义了 Roman 字体，则将使用它代替其它字体。

映射部分

`mp.conf` 文件的映射部分从中间代码范围映射到语言环境中的字体组。 此部分中的每一行都具有如下形式。

关键字 MapCode2Font。

range_start 以 `0x` 开头的 4–字节十六进制值表示要映射到一个或多个字体组的代码范围的开始。

range_end 表示要映射的代码范围的结尾。它可以是 '-' 值;其中只有单个中间代码点被映射到目标字体。

组 一个 Type1、PCF 或 TrueType 字体组，表示形式通过它进行打印。

关联部分

`mp.conf` 文件的关联部分将每种字体与共享对象相关联，此共享对象把中间代码点映射到字体编码中的变形显现形式。此部分中的每一行都具有如下形式。

关键字 `CnvCode2Font`。

字体别名 为字体定义的别名。

映射函数	获取中间代码并返回字体编码中的变形显现形式，变形显现形式再获取字型索引，从而得到字型。
具有映射函数的文件路径	包含映射函数的 .so 文件名。您可以使用 dumpcs 中的公用程序来查找 EUC 语言环境的中间代码集。

注意：由 mp (1) 使用的当前 TrueType 引擎只能处理格式 4 和 PlatformID 3 cmap。即，您只能配置 Microsoft .ttf 文件。 另外，要想使 TrueType 字体正常工作，字符映射编码必须是 Unicode 或 Symbol。由于 Solaris 环境中的大多数 .ttf 字体遵循这些限制，因此您可以在 mp.conf 文件内映射 Solaris 软件中的所有 TrueType 字体。

您创建用于映射与 PCF type1 X 逻辑字体说明 (XLFD) 相对应的字体的共享对象后，再创建从中间代码范围映射到由 XLFD 指定的编码的共享对象。 例如：

```
-monotype-arial-bold-r-normal-bitmap-10-100-75-75-p-54-iso8859-8
```

相应的 PCF 字体为：

```
/usr/openwin/lib/locale/iso_8859_8/X11/fonts/75dpi/ariabd10.pcf.Z
```

该字体以 isoISO 8859-8 进行编码，因此共享对象必须在中间代码和对应的 ISO 8859-8 代码点之间映射。

如果带有 XLFD 的 TrueType 字体：

```
-monotype-arial-medium-r-normal--0-0-0-0-p-0-iso8859-8
```

具有对应的字体：

```
/usr/openwin/lib/locale/iso_8859_8/X11/fonts/TrueType/arial__h.ttf
```

您应该在中间代码和 Unicode 之间映射，因为前一种 TrueType 字体的 cmap 编码在 Unicode 中。在此 TrueType 字体示例中，假定 en_US.UTF-8 语言环境中与希伯来字符（由 PLS 层产生）相对应的样例中间代码为 0xe50000e9。 由于该字体为 Unicode 编码，因此以这样的方式来设计对应的 .so 模块内的函数，即当您传递 0xe50000e9时，输出与 Unicode 中的变形显现形式相对应。 此处的示例为 0x000005d9。

映射函数的函数原型应为：

```
unsigned int function(unsigned int inter_code_pt)
```

以下是可在 mp.conf：中使用的可选关键字/值对：

```
PresentationForm WC/PLSOutput
```

缺省值为 PLSOutput。如果用户指定 WC，则生成的中间代码点为宽字符。 对于 CTL 打印，应使用此缺省值。

如果语言环境为非 CTL 语言环境并且键盘值为 PLSOutput，则该值将被忽略，而且 mp (1) 将生成宽字符代码。

如果语言环境支持 CTL，则您可以使用下表中列出的可选关键字/值对。这些变量可以假设该表中间列中给出的任何可能值。

表 7–1 可选关键字/值对

可选关键字	可选值	缺省
Orientation	ORIENTATION_LTR/ ORIENTATION_RTL/ ORIENTATION_CONTEXTUAL	ORIENTATION_LTR
Numerals	NUMERALS_NOMINAL/ NUMERALS_NATIONAL/ NUMERALS_CONTEXTUAL	NUMERALS_NOMINAL
TextShaping	TEXT_SHAPED/ TEXT_NOMINAL/ TEXT_SHFORM1/ TEXT_SHFORM2/ TEXT_SHFORM3/ TEXT_SHFORM4	TEXT_SHAPED

添加打印机常驻字体

下列示例说明了将新的 PCF、TrueType 或 Type1 打印机常驻字体添加到配置文件时需要遵循的步骤。

将用于显示 0x00000021 - 0x0000007f 范围内字符的字体替换为 TrueType 字体，而不是当前配置的 PCF 字体。

在添加新字体前，请查看配置文件中与当前配置字体相对应的各种组件，如下所示。

```
FontNameAlias iso88591R  PCF  /usr/openwin/lib/X11/fonts/75dpi/courR18PCF.Z
FontNameAlias iso88591B  PCF  /usr/openwin/lib/X11/fonts/75dpi/courB18PCF.Z
.
.
.
FontGroup       iso88591         PCF       iso88591R iso88591B
.
.
.
MapCode2Font    0x00000020      0x0000007f      iso88591
.
.
```

```
.
CnvCode2Font iso88591R _xuiso88591 /usr/lib/lp/locale/$LANG/mp/xuiso88591.so
CnvCode2Font iso88591B _xuiso88591 /usr/lib/lp/locale/$LANG/mp/xuiso88591.so
```

假设您选择了 /usr/openwin/lib/locale/ja/X11/fonts/TT/HG-MinchoL.ttf 作为在 en_US.UTF-8 语言环境中进行映射的候选文件。由于这是一个 Unicode 字符映射的 TrueType 字体文件，因此在 .so 内的映射函数中，您只需要有直接返回传入 ucs-2 代码点的函数。

```
unsigned short _ttfjis0201(unsigned short ucs2) {
                 return(ucs2);
         }
```

将此程序代码保存在 ttfjis0201.c 文件中。 按如下方式创建一个共享对象。

```
cc -G -Kpic -o ttfjis0201.so ttfjis0201.c
```

但是，如果您正在映射 PCF 文件（如 /usr/openwin/lib/locale/ja/X11/fonts/75dpi/gotmrk20.pcf.Z ），则在 /usr/openwin/lib/locale/ja/X11/fonts/75dpi/ 目录中查找 fonts.dir 文件。熟悉与 XLFD 相对应的编码，即：

```
-sun-gothic-medium-r-normal--22-200-75-75-c-100-jisx0201.1976-0
```

如果 jisx0201 为该编码，请准备一个从 ucs-2 映射到 jisx0201 的共享对象。您需要获得映射表，以便创建 .so 模块（如果一个都没有提供的话）。对于 Unicode 语言环境，查找从许多字符集到 ftp.unicode.org/pub/MAPPINGS/ 下的Unicode 的映射。请遵循这些映射 (1)(1) 以编写 xu2jis0201.c 文件：

```
 unsigned short _xu2jis0201(unsigned short ucs2) {
                         if(ucs2>= 0x20 && ucs2 <= 0x7d )
                                 return (ucs2);
                         if(ucs2==0x203e)
                                 return (0x7e);
                         if(ucs2>= 0xff61 && ucs2 <= 0xff9f)
                                 return (ucs2 - 0xff60 + 0xa0);
                        return(0);
                 }
```

当您创建一个映射文件时，请将所有 UCS-2 包括到 jisx0201 事例中。

```
cc  -G -o xu2jis0201.so xu2jis0201.c
```

创建共享对象文件

此示例创建一个共享对象文件。

添加此字体的方法是：将下面的行添加到 mp.conf 文件的相应部分。下列示例说明如何添加 TrueType 字体。除了更改 PCF 关键字（而不是TrueType）外， PCF 字体遵循同一模式。

```
FontNameAlias   jis0201R TrueType /home/fn/HG-Minchol.ttf
FontGroup     jis0201 TrueType jis0201R
```

```
MapCode2Font   0x0020      0x007f   jis0201
CnvCode2Font    jis0201R       _ttfjis0201 <.so path>
```

其中 .so 路径指向 xu2jis0201.so 文件。

用已更改的 mp.conf 文件调用 mp(1) 会导致范围 0x0020-0x007f 以新字体打印。也用同样的 .so 文件映射其它日文字符范围，例如，范围 0x0000FF61 0x0000FF9F。

为了维护向下兼容性， /usr/openwin/lib/locale/$LANG/print/prolog.ps 文件（如果存在的话）用于在当前语言环境（其中 $LANG 是语言环境组件之一）中创建输出。在这种情况下，不使用任何配置文件机制。

请参阅 /usr/lib/lp/locale/en_US.UTF-8/mp/mp.conf，它是一个样例 mp.conf 文件。

定制现有的 prolog 文件并添加新的 prolog 文件

prolog 文件可分为两种主要类别：

- PostScript prolog 文件（.ps）
- X 打印服务器客户程序 prolog 文件（.xpr）。

PostScript 文件定制

PostScript 文件属于下列类别：

- 通用 prolog 文件
- 打印布局 prolog 文件

依赖于语言环境的 prolog 文件

prolog.ps 文件的目的是设置非通用字体。应用程序使用这些预定义 PostScript 字体名称进行打印。prolog 文件必须为桌面设置日历管理器和 mp 至少定义下面字体名称。

- LC_Times-Roman
- LC_Times-Bold
- LC_Helvetica
- LC_Helvetica-Bold
- LC_Courier

- LC_Helvetica-BoldOblique
- LC_Times-Italic

下列示例使用这些字体打印指定的特殊本地字符集：

```
100 100 moveto
/LC_Times-Roman findfont 24 scale font setfont（您的语言环境中的任何字符串）显示
```

Solaris 本地化工具包为日语环境提供了一个样例 `prolog.ps` 文件。另外，此文件也可以在 `/usr/openwin/lib/locale/ja/print/` 目录中找到。

下列示例说明如何在现有的 `prolog.ps` 中添加或更改复合字体。

```
%
(Foo-Fine) makecodeset12
(Base-Font) makeEUCfont
%
```

假设您要定义一种称为 LC_Base-Font 的复合字体：

LC_Base-Font 是 Foo-Fine 和名为 Base-Font 的基准字体的复合字体。Foo-Fine 是包含本地字符集的字体。您不需要任何高深的 PostScript 知识就可以添加或更改字体。

对示例版本进行研究是创建 `prolog.ps` 文件的最好方法。在示例 `prolog.ps` 中，需要编写两个例程，即 `makecodeset12` 和 `makeEUCfont`。 `Makecodeset12` 设置本地字体编码信息。此例程可能会因语言环境的不同而不同。`MakeEUCfont` 将基准字体和区域字体合并为一个复合字体。`prolog` 文件的创建者应具有良好的 PostScript 知识，以便编写 `makecodeset12` 和 `makeEUCfont`。

`prolog.ps` 文件支持只保持向后兼容性。不要为一种语言环境的打印需要创建新的 `prolog.ps` 文件。请换而使用 `mp.conf`。

`prolog.ps` 的路径为

```
/usr/openwin/lib/locale/$LANG/print/prolog.ps
```

通用 PostScript `prolog` 文件

通用 `prolog` 文件为 `mp.common.ps` 。

其它每个页面布局 prolog 文件都需要包含此文件。

`mp.common.ps` 文件位于 `/usr/lib/lp/locale/C/mp/` 目录中，它包含一个 PostScript 例程以便将标准编码中的字体重新编码为 ISO 8859–1 编码。`.reencodeISO` 例程从打印布局 `prolog` 文件中调用以更改字体的编码。通常情况下，此 `prolog` 文件不需要任何定制。如果用户正在创建自己的 `prolog` 文件，请将环境变量 `MP_PROLOGUE` 设置为指向包含修改后的 `prolog` 文件的目录。

打印布局 prolog 文件

打印布局 prolog 文件、mp.*.ps 文件包含控制用于打印的页面布局的例程。除了为具有用户名、打印日期和页号的打印页提供标题和页脚外，这些 prolog 文件还可以提供其它信息。例如，prolog 文件可提供有效的打印区域尺寸以及打印所要使用的横向和纵向模式。

打印布局 prolog 文件为：

- mp.pro.ps
- mp.pro.alt.ps
- mp.pro.fp.ps
- mp.pro.ps
- mp.pro.ts.ps
- mp.pro.altl.ps
- mp.pro.ff.ps
- mp.pro.l.ps
- mp.pro.ll.ps
- mp.pro.tm.ps

需要在每个 prolog 文件中定义一组标准函数。当新的打印页启动、某个打印页结束或新的列结束时这些函数即被调用。这些函数的实现定义了打印输出的打印属性。

下列 PostScript 变量被 mp(1) 程序在运行时定义。所有的打印布局文件都可以使用这些变量来打印动态信息，如 用户名、 主题 和 打印时间。 从变量获取的此信息通常出现在打印页的标题或页脚中。

用户 正在运行 mp 的用户的名称，可从系统 passwd 文件中获取。

MailFor 用于保存要打印的文章类型名称的变量。此变量的可能值为：

- "Listing for" - 当输入为文本文件时
- "Mail for" - 当输入为邮件文件时
- "Article from" - 当输入为新闻组文章时

主题 从邮件和新闻标题中获取的主题。您可以使用 -s 选项将某个主题强迫为邮件和新闻文件以及正常的文本文件。

Timenow 出现在标题和页脚中的打印时间。此信息从 localtime（）函数中获取。

下列函数在打印布局 prolog 文件中实现。所有这些函数都可以使用子函数。

endpage 用法：page_number endpage

当到达打印页的底部时调用。此函数恢复页面的图形上下文并发出 "showpage" 命令。在某些 prolog 文件中，标题和页脚信息仅以逐页模式（而不是逐列模式）显示。您可以实现此函数以调用显示标题和页脚灰度级菱形的子函数。

newpage	用法： page_number newpage 当某个新页开始时要执行的例程或命令。设置横向打印模式、保存打印图形上下文和翻译页坐标是用于例程的一些函数。
endcol	用法： page_number col_number endcol 用于显示标题和页脚信息。移到新的打印位置，等等。

要添加新的打印布局 prolog 文件，您需要在打印布局 prolog 文件内显式定义下列变量。

NumCols 打印页中的列数。缺省值为 2。

PrintWidth 打印区域的宽度（以英寸为单位）。缺省值为 6。

PrintHeight 打印区域的高度（以英寸为单位）。缺省值为 9。

.xpr 文件定制

默认情况下，这些文件位于 /usr/lib/lp/locale/C/mp/ 。.xpr 文件与每个 PostScript prolog 布局文件相对应，mp.common.ps 除外。您可以通过定义 MP_PROLOGUE 环境变量来定义替换的 prolog 目录。

这些文件用作关键字/值对。以 # 开头的行被认为是注释。空格分隔不同的标记（除非显式进行说明）。每个 .xpr 文件的三个主要部分由下列关键字对绑定：

- STARTCOMMON/ENDCOMMON
- STARTPAGE/ENDPAGE
- STARTCOLUMN/ENDCOLUMN
- STARTFORCEDPAGE/ENDFORCEDPAGE
- STARTFORCEDCOLUMN/ENDFORCEDCOLUMN

某些关键字/值对可用于这三个区域。下面分别描述每个区域。

STARTCOMMON/ENDCOMMON 关键字

出现在 STARTCOMMON 关键字之后并且在 ENDCOMMON 关键字之前的所有关键字/值对定义打印页的常规属性。通过使用“/”来分隔关键字的不同有效值。

ORIENTATION 0/1
“0”表示以纵向模式打印而“1”表示以横向模式打印。

PAGELENGTH **无符号整数**
一个值，用于表示每个逻辑页的行数。

LINELENGTH **无符号整数**
一个值，用于表示每行的单列字符数。

NUMCOLS **无符号整数**
每个物理页的逻辑页数。

HDNGFONTSIZE **无符号整数**
标题字体磅值（以小数点表示）。

BODYFONTSIZE **无符号整数**
正文字体磅值（以小数点表示）。

PROLOGDPI **无符号整数**
创建当前 .xpr 文件使用的每英寸点数比例。

YTEXTBOUNDARY **无符号整数**
此 y 坐标确定了页或逻辑页（列）中文本打印的边界。此边界还可以用于检查文本打印是否发生在预期的区域内。在复杂文本布局和 EUC 打印时需要此边界，这是因为从相应的字体中获取的字符高度信息有可能是错的。

STARTTEXT **无符号整数 无符号整数**
以小数点表示的 x/y 点，这是物理页的第一个逻辑页中实际文本打印的开始位置。

PAGESTRING 0/1
1 表示需要在标题中的页号前附加一个“Page”字符串。

0 表示只显示页号。

EXTRAHDNGFONT *font string 1*、 *font string 2*、*... font string n*
“font string 1”到“font string n”是 X 逻辑字体描述。将关键字 EXTRAHDNGFONT 从逗号分隔字体名称列表中分隔开的标记是 "，而不是空格或制表符。 当打印标题时，给定的首选项是这些字体，而不是内置字体。通常情况下，EXTRABODYFONT 用于分配在 /usr/openwin/server/etc/XpConfig/C/print/models/<model name>/fonts 目录中配置的打印机常驻字体。

fonts.dir 文件包含打印机常驻字体的 XLFD。

一种字体通常被指定为

```
"-monotype-Gill Sans-Regular-r-normal- -*-%d-*-*-p-0-iso8859-2"
```

在 .xpr 文件中。 "%d（如果存在的话）被 mp(1) 替换为 .xpr 文件中当前标题字体的磅值。x 分辨率和 y 分辨率由 * 指定，并且平均宽度字段设置为 0 以表示可缩放字体的选择（如果可能的话）。您也可以给出更多的特定字体名称。

EXTRABODYFONT *font string 1*、*font string 2*、*... font string n*
与 EXTRAHDNGFONT 相同，除了这些字体被用来打印页面正文以外。

XDISPLACEMENT **带符号/无符号** *int*
给出要应用于该页的 x 坐标位移，以便在 x 方向移动该页的内容。此位移可以是 +ve 或 -ve 值。

YDISPLACEMENT *signed/unsigned int*
与 x 位移相同（除了移动发生在 y 方向以外）。

当您发现某些打印机具有非标准页边距宽度并且需要在页中移动打印内容时，这两个关键字很有用。

STARTPAGE/ENDPAGE

此部分中的关键字值对由 STARTPAGE 和 ENDPAGE 关键字绑定。此部分包含要用于物理页的绘图和标题信息。物理页可包含许多逻辑页，但是，包含在这些关键字之间的所有绘图例程仅一次应用于逻辑页。

有效的绘图实体为 LINE 和 ARC。XDrawLine() 和 XDrawArc() 函数在这些关键字的值上执行。

此部分内的尺寸被映射在 PROLOGDPI 单元中。角度以度数表示。

LINE x1 y1 x2 y2	x/y 无符号坐标定义连接一条直线的两个点。
ARC x y width height angle1 angle2	x 和 y 均为表示弧线起始点的无符号整数。width 和 height 为表示弧线的宽度和高度的无符号整数。
USERSTRINGPOS x y	无符号坐标表示标题中用户信息的打印位置。
TIMESTRINGPOS x y	无符号坐标表示标题中打印时间的打印位置。
PAGESTRINGPOS x y	无符号坐标表示每个打印页的页字符串的打印位置。
SUBJECTSTRINGPOS x y	无符号坐标表示页中主题的打印位置。

STARTFORCEDPAGE/ENDFORCEDPAGE

当 -n 选项分配给 mp 时，STARTPAGE/ENDPAGE 部分内的所有装饰都不打印。但是，即使给出 -n 选项，也会打印 STARTFORCEDPAGE/ENDFORCEDPAGE 部分内所包含的任何内容。

STARTCOLUMN/ENDCOLUMN

所有关键字与 第 154 页“STARTPAGE/ENDPAGE” 中描述的相同，除了此部分中的条目可 NUMCOLS 次应用于物理页以外。

如果 NUMCOLS 为 3，则物理页的可打印区域将被分为三个部分，并且每页中直线、弧线或标题装饰将出现三次。

STARTFORCEDCOLUMN/ENDFORCEDCOLUMN

当 -n 选项分配给 mp 时，STARTCOLUMN/ENDCOLUMN 部分内给定的所有装饰都不打印。但是，即使给出 -n 选项，也会打印 STARTFORCEDCOLUMN/ENDFORCEDCOLUMN 部分内包含的任何内容。

创建新的 .xpr 文件

下面的值是针对不同关键字的 mp(1) 程序缺省值（如果在 STARTCOMMON/ENDCOMMON 部分的 .xpr 文件中没有指定这些值的话）。

```
ORIENTATION 0
PAGELENGTH 60
LINELENGTH 80
YTEXTBOUNDARY 3005
NUMCOLS 01
HDNGFONTSIZE 120
PROLOGDPI 300
STARTTEXT 135 280
PAGESTRING 0
```

由 STARTPAGE/ENDPAGE 和 STARTCOLUMN/ENDCOLUMN 绑定的其它两个部分不需要缺省值。

当创建新的 .xpr prolog 文件时，只需要指定与缺省值不同的值。

若要创建不带装饰的页，请使用如下方式：每个物理页有四个逻辑页，打印格式为纵向格式。

- STARTCOMMON
- NUMCOLS 04
- LINELENGTH 20
- ENDCOMMON

在这种情况下，您不需要其它两个部分：

- STARTPAGE/ENDPAGE
- STARTCOLUMN/ENDCOLUMN

如果您不在打印页中加入装饰，则不需要这些参数。所有坐标的缺省值均为 300 dpi，除非您没有指定 PROLOGDPI 关键字。如果目标打印机的分辨率不同，程序将调整 .xpr 文件以适应该分辨率。

当创建 .xpr 文件时，您必须事先知道纸张的尺寸。在美国使用的纸张为 8.5x11 英寸，对于分辨率为 300 dpi 的打印机，2550X3300 为总的尺寸。大多数打印机无法从纸张的左上角打印。相反，它们在物理页周围留一定的页边距。这意味着即使您试图从 0,0 打印，也不会打印到页的左上角。当创建新的 .xpr 文件时，您需要考虑此限制。

附录 A

iconv 代码转换

下表列出了 Solaris 9 环境中与 Unicode 相关的可用代码转换模块。

表 A–1 与 Unicode 相关的 `iconv` 可用代码转换模块

从代码（符号）	到代码（符号）
646 (ISO 646)	UCS-2
646 (ISO 646)	USC-2BE
646 (ISO 646)	UCS-2LE
646 (ISO 646)	USC-4
646 (ISO 646)	USC-4BE
646 (ISO 646)	USC-4LE
646 (ISO 646)	UTF-8
646 (ISO 646)	UTF-16
646 (ISO 646)	UTF-16BE
646 (ISO 646)	UTF-16LE
646 (ISO 646)	UTF-32
646 (ISO 646)	UTF-32BE
646 (ISO 646)	UTF-32LE
ISO8859–11	UTF-8
8859-1 (ISO8859-1)	UCS-2
8859-1 (ISO8859-1)	UCS-2BE
8859-1 (ISO8859-1)	UCS-2LE

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
8859-1 (ISO8859-1)	UCS-4
8859-1 (ISO8859-1)	UCS–4BE
8859-1 (ISO8859-1)	UCS-4LE
8859-1 (ISO8859-1)	UTF-8
8859-1 (ISO8859-1)	UTF-16
8859-1 (ISO8859-1)	UTF-16BE
8859-1 (ISO8859-1)	UTF-16LE
8859-1 (ISO8859-1)	UTF-32
8859-1 (ISO8859-1)	UTF-32BE
8859-1 (ISO8859-1)	UTF-32LE
8859-2 (ISO8859-2)	UCS-2
8859-2 (ISO8859-2)	UCS-2BE
8859-2 (ISO8859-2)	UCS-2LE
8859-2 (ISO8859-2)	UCS-4
8859-2 (ISO8859-2)	UCS–4BE
8859-2 (ISO8859-2)	UCS-4LE
8859-2 (ISO8859-2)	UTF-8
8859-2 (ISO8859-2)	UTF-16
8859-2 (ISO8859-2)	UTF-16BE
8859-2 (ISO8859-2)	UTF-16LE
8859-2 (ISO8859-2)	UTF-32
8859-2 (ISO8859-2)	UTF-32BE
8859-2 (ISO8859-2)	UTF-32LE
8859-3 (ISO8859-3)	UCS-2
8859-3 (ISO8859-3)	UCS-2BE
8859-3 (ISO8859-3)	UCS-2LE
8859-3 (ISO8859-3)	UCS-4
8859-3 (ISO8859-3)	UCS–4BE

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块　（续下）

从代码（符号）	到代码（符号）
8859-3 (ISO8859-3)	UCS-4LE
8859-3 (ISO8859-3)	UTF-8
8859-3 (ISO8859-3)	UTF-16
8859-3 (ISO8859-3)	UTF-16BE
8859-3 (ISO8859-3)	UTF-16LE
8859-3 (ISO8859-3)	UTF-32
8859-3 (ISO8859-3)	UTF-32BE
8859-3 (ISO8859-3)	UTF-32LE
8859-4 (ISO8859-4)	UCS-2
8859-4 (ISO8859-4)	UCS-2BE
8859-4 (ISO8859-4)	UCS-2LE
8859-4 (ISO8859-4)	UCS-4
8859-4 (ISO8859-4)	UCS–4BE
8859-4 (ISO8859-4)	UCS-4LE
8859-4 (ISO8859-4)	UTF-8
8859-4 (ISO8859-4)	UTF-16
8859-4 (ISO8859-4)	UTF-16BE
8859-4 (ISO8859-4)	UTF-16LE
8859-4 (ISO8859-4)	UTF-32
8859-4 (ISO8859-4)	UTF-32BE
8859-4 (ISO8859-4)	UTF-32LE
8859-5 (ISO8859-5)	UCS-2
8859-5 (ISO8859-5)	UCS-2BE
8859-5 (ISO8859-5)	UCS-2LE
8859-5 (ISO8859-5)	UCS-4
8859-5 (ISO8859-5)	UCS–4BE
8859-5 (ISO8859-5)	UCS-4LE
8859-5 (ISO8859-5)	UTF-8

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
8859-5 (ISO8859-5)	UTF-16
8859-5 (ISO8859-5)	UTF-16BE
8859-5 (ISO8859-5)	UTF-16LE
8859-5 (ISO8859-5)	UTF-32
8859-5 (ISO8859-5)	UTF-32BE
8859-5 (ISO8859-5)	UTF-32LE
8859-6 (ISO8859-6)	UCS-2
8859-6 (ISO8859-6)	UCS-2BE
8859-6 (ISO8859-6)	UCS-2LE
8859-6 (ISO8859-6)	UCS-4
8859-6 (ISO8859-6)	UCS–4BE
8859-6 (ISO8859-6)	UCS-4LE
8859-6 (ISO8859-6)	UTF-8
8859-6 (ISO8859-6)	UTF-16
8859-6 (ISO8859-6)	UTF-16BE
8859-6 (ISO8859-6)	UTF-16LE
8859-6 (ISO8859-6)	UTF-32
8859-6 (ISO8859-6)	UTF-32BE
8859-6 (ISO8859-6)	UTF-32LE
8859-7 (ISO8859-7)	UCS-2
8859-7 (ISO8859-7)	UCS-2BE
8859-7 (ISO8859-7)	UCS-2LE
8859-7 (ISO8859-7)	UCS-4
8859-7 (ISO8859-7)	UCS–4BE
8859-7 (ISO8859-7)	UCS-4LE
8859-7 (ISO8859-7)	UTF-8
8859-7 (ISO8859-7)	UTF-16
8859-7 (ISO8859-7)	UTF-16BE

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
8859-7 (ISO8859-7)	UTF-16LE
8859-7 (ISO8859-7)	UTF-32
8859-7 (ISO8859-7)	UTF-32BE
8859-7 (ISO8859-7)	UTF-32LE
8859-8 (ISO8859-8)	UCS-2
8859-8 (ISO8859-8)	UCS-2BE
8859-8 (ISO8859-8)	UCS-2LE
8859-8 (ISO8859-8)	UCS-4
8859-8 (ISO8859-8)	UCS–4BE
8859-8 (ISO8859-8)	UCS-4LE
8859-8 (ISO8859-8)	UTF-8
8859-8 (ISO8859-8)	UTF-16
8859-8 (ISO8859-8)	UTF-16BE
8859-8 (ISO8859-8)	UTF-16LE
8859-8 (ISO8859-8)	UTF-32
8859-8 (ISO8859-8)	UTF-32BE
8859-8 (ISO8859-8)	UTF-32LE
8859-9 (ISO8859-9)	UCS-2
8859-9 (ISO8859-9)	UCS-2BE
8859-9 (ISO8859-9)	UCS-2LE
8859-9 (ISO8859-9)	UCS-4
8859-9 (ISO8859-9)	UCS–4BE
8859-9 (ISO8859-9)	UCS-4LE
8859-9 (ISO8859-9)	UTF-8
8859-9 (ISO8859-9)	UTF-16
8859-9 (ISO8859-9)	UTF-16BE
8859-9 (ISO8859-9)	UTF-16LE
8859-9 (ISO8859-9)	UTF-32

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
8859-9 (ISO8859-9)	UTF-32BE
8859-9 (ISO8859-9)	UTF-32LE
8859-10 (ISO8859-10)	UCS-2
8859-10 (ISO8859-10)	UCS-2BE
8859-10 (ISO8859-10)	UCS-2LE
8859-10 (ISO8859-10)	UCS-4
8859-10 (ISO8859-10)	UCS–4BE
8859-10 (ISO8859-10)	UCS-4LE
8859-10 (ISO8859-10)	UTF-8
8859-10 (ISO8859-10)	UTF-16
8859-10 (ISO8859-10)	UTF-16BE
8859-10 (ISO8859-10)	UTF-16LE
8859-10 (ISO8859-10)	UTF-32
8859-10 (ISO8859-10)	UTF-32BE
8859-10 (ISO8859-10)	UTF-32LE
8859-13 (ISO8859-13)	UCS-2
8859-13 (ISO8859-13)	UCS-2BE
8859-13 (ISO8859-13)	UCS-2LE
8859-13 (ISO8859-13)	UCS-4
8859-13 (ISO8859-13)	UCS–4BE
8859-13 (ISO8859-13)	UCS-4LE
8859-13 (ISO8859-13)	UTF-8
8859-13 (ISO8859-13)	UTF-16
8859-13 (ISO8859-13)	UTF-16BE
8859-13 (ISO8859-13)	UTF-16LE
8859-13 (ISO8859-13)	UTF-32
8859-13 (ISO8859-13)	UTF-32BE
8859-13 (ISO8859-13)	UTF-32LE

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
8859-14 (ISO8859-14)	UCS-2
8859-14 (ISO8859-14)	UCS-2BE
8859-14 (ISO8859-14)	UCS-2LE
8859-14 (ISO8859-14)	UCS-4
8859-14 (ISO8859-14)	UCS–4BE
8859-14 (ISO8859-14)	UCS-4LE
8859-14 (ISO8859-14)	UTF-8
8859-14 (ISO8859-14)	UTF-16
8859-14 (ISO8859-14)	UTF-16BE
8859-14 (ISO8859-14)	UTF-16LE
8859-14 (ISO8859-14)	UTF-32
8859-14 (ISO8859-14)	UTF-32BE
8859-14 (ISO8859-14)	UTF-32LE
8859-15 (ISO8859-15)	UCS-2
8859-15 (ISO8859-15)	UCS-2BE
8859-15 (ISO8859-15)	UCS-2LE
8859-15 (ISO8859-15)	UCS-4
8859-15 (ISO8859-15)	UCS–4BE
8859-15 (ISO8859-15)	UCS-4LE
8859-15 (ISO8859-15)	UTF-8
8859-15 (ISO8859-15)	UTF-16
8859-15 (ISO8859-15)	UTF-16BE
8859-15 (ISO8859-15)	UTF-16LE
8859-15 (ISO8859-15)	UTF-32
8859-15 (ISO8859-15)	UTF-32BE
8859-15 (ISO8859-15)	UTF-32LE
8859-16 (ISO8859-16)	UCS-2
8859-16 (ISO8859-16)	UCS-2BE

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
8859-16 (ISO8859-16)	UCS-2LE
8859-16 (ISO8859-16)	UCS-4
8859-16 (ISO8859-16)	UCS–4BE
8859-16 (ISO8859-16)	UCS-4LE
8859-16 (ISO8859-16)	UTF-8
8859-16 (ISO8859-16)	UTF-16
8859-16 (ISO8859-16)	UTF-16BE
8859-16 (ISO8859-16)	UTF-16LE
8859-16 (ISO8859-16)	UTF-32
8859-16 (ISO8859-16)	UTF-32BE
8859-16 (ISO8859-16)	UTF-32LE
eucJP	UTF-8
gb2312	UTF-8
iso2022	UTF-8
ko_KR-cp933	UTF-8
ko_KR-euc	UTF-8
ko_KR-iso2022–7	UTF-8
ko_KR-johap	UTF-8
ko_KR-johap92	UTF-8
zh_TW-euc	UTF-8
zh_TW-cp937	UTF-8
zh_TW-iso2022–7	UTF-8
GBK	UTF-8
FujitsuJEF-ascii-code	UTF-8
FujitsuJEF-ascii-face	UTF-8
FujitsuJEF-kana-code	UTF-8
FujitsuJEF-kana-face	UTF-8
HitachiKEIS83	UTF-8

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
HitachiKEIS90	UTF-8
ISO-2022–JP	UTF-8
KOI8-R	UCS-2
KOI8-R	UCS-2BE
KOI8-R	UCS-2LE
KOI8-R	UCS-4
KOI8-R	UCS–4BE
KOI8-R	UCS-4LE
KOI8-R	UTF-8
KOI8-R	UTF-16
KOI8-R	UTF-16BE
KOI8-R	UTF-16LE
KOI8-R	UTF-32
KOI8-R	UTF-32BE
KOI8-R	UTF-32LE
KOI8-U	UCS-2
KOI8-U	UCS-2BE
KOI8-U	UCS-2LE
KOI8-U	UCS-4
KOI8-U	UCS-4BE
KOI8-U	UCS-4LE
KOI8-U	UTF–8
KOI8-U	UTF-16
KOI8-U	UTF-16BE
KOI8-U	UTF-16LE
KOI8-U	UTF-32
KOI8-U	UTF-32BE
KOI8-U	UTF-32LE

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
NECJIPS	UTF-8
PCK	UTF-8
UCS-2	646 (ISO 646)
UCS-2	8859-1 (ISO8859-1)
UCS-2	8859-2 (ISO8859-2)
UCS-2	8859-3 (ISO8859-3)
UCS-2	8859-4 (ISO8859-4)
UCS-2	8859-5 (ISO8859-5)
UCS-2	8859-6 (ISO8859-6)
UCS-2	8859-7 (ISO8859-7)
UCS-2	8859-8 (ISO8859-8)
UCS-2	8859-9 (ISO8859-9)
UCS-2	8859-10 (ISO8859-10)
UCS-2	8859-13 (ISO8859-13)
UCS-2	8859-14 (ISO8859-14)
UCS-2	8859-15 (ISO8859-15)
UCS-2	8859-16 (ISO8859-16)
UCS-2	KOI8-R
UCS-2	KOI8-U
UCS-2	UCS-4
UCS-2	UCS-4BE
UCS-2	UCS-4LE
UCS-2	UTF-7
UCS-2	UTF-8
UCS-2BE	646 (ISO 646)
UCS-2BE	8859-1 (ISO8859-1)
UCS-2BE	8859-2 (ISO8859-2)
UCS-2BE	8859-3 (ISO8859-3)

表 A–1 与 Unicode 相关的 `iconv` 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UCS-2BE	8859-4 (ISO8859-4)
UCS-2BE	8859-5 (ISO8859-5)
UCS-2BE	8859-6 (ISO8859-6)
UCS-2BE	8859-7 (ISO8859-7)
UCS-2BE	8859-8 (ISO8859-8)
UCS-2BE	8859-9 (ISO8859-9)
UCS-2BE	8859-10 (ISO8859-10)
UCS-2BE	8859-13 (ISO8859-13)
UCS-2BE	8859-14 (ISO8859-14)
UCS-2BE	8859-15 (ISO8859-15)
UCS-2BE	8859-16 (ISO8859-16)
UCS-2BE	KOI8-R
UCS-2BE	KOI8-U
UCS-2BE	UCS-4
UCS-2BE	UCS-4BE
UCS-2BE	UCS-4LE
UCS-2BE	UTF-8
UCS-2LE	646 (ISO 646)
UCS-2LE	8859-1 (ISO8859-1)
UCS-2LE	8859-2 (ISO8859-2)
UCS-2LE	8859-3 (ISO8859-3)
UCS-2LE	8859-4 (ISO8859-4)
UCS-2LE	8859-5 (ISO8859-5)
UCS-2LE	8859-6 (ISO8859-6)
UCS-2LE	8859-7 (ISO8859-7)
UCS-2LE	8859-8 (ISO8859-8)
UCS-2LE	8859-9 (ISO8859-9)
UCS-2LE	8859-10 (ISO8859-10)

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 *(续下)*

从代码（符号）	到代码（符号）
UCS-2LE	8859-13 (ISO8859-13)
UCS-2LE	8859-14 (ISO8859-14)
UCS-2LE	8859-15 (ISO8859-15)
UCS-2LE	8859-16 (ISO8859-16)
UCS-2LE	KOI8-R
UCS-2LE	KOI8-U
UCS-2LE	UCS-4
UCS-2LE	UCS-4BE
UCS-2LE	UCS-4LE
UCS-2LE	UTF-8
UCS-2LE	UTF-32
UCS-2LE	UTF-32BE
UCS-2LE	UTF-32LE
UCS-4	646
UCS-4	8859-1 (ISO8859-1)
UCS-4	8859-2 (ISO8859-2)
UCS-4	8859-3 (ISO8859-3)
UCS-4	8859-4 (ISO8859-4)
UCS-4	8859-5 (ISO8859-5)
UCS-4	8859-6 (ISO8859-6)
UCS-4	8859-7 (ISO8859-7)
UCS-4	8859-8 (SO 8859-8)
UCS-4	8859-9 (ISO8859-9)
UCS-4	8859-10 (ISO8859-10)
UCS-4	8859-13 (ISO8859-13)
UCS-4	8859-14 (ISO8859-14)
UCS-4	8859-15 (ISO8859-15)
UCS-4	8859-16 (ISO8859-16)

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UCS-4	KOI8-R
UCS-4	KOI8-U
UCS-4	UCS-2
UCS-4	UCS-2BE
UCS-4	UCS-2LE
UCS-4	UTF-7
UCS-4	UTF-8
UCS-4	UCS-16
UCS-4	UCS-16BE
UCS-4	UCS-16LE
UCS-4	UTF-32
UCS-4	UCS-32BE
UCS-4	UCS-32LE
UCS-4BE	646
UCS-4BE	8859-1 (ISO8859-1)
UCS-4BE	8859-2 (ISO8859-2)
UCS-4BE	8859-3 (ISO8859-3)
UCS-4BE	8859-4 (ISO8859-4)
UCS-4BE	8859-5 (ISO8859-5)
UCS-4BE	8859-6 (ISO8859-6)
UCS-4BE	8859-7 (ISO8859-7)
UCS-4BE	8859-8 (SO 8859-8)
UCS-4BE	8859-9 (ISO8859-9)
UCS-4BE	8859-10 (ISO8859-10)
UCS-4BE	8859-13 (ISO8859-13)
UCS-4BE	8859-14 (ISO8859-14)
UCS-4BE	8859-15 (ISO8859-15)
UCS-4BE	8859-16 (ISO8859-16)

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UCS-4BE	KOI8-R
UCS-4BE	KOI8-U
UCS-4BE	UCS-2
UCS-4BE	UCS-2BE
UCS-4BE	UCS-2LE
UCS-4BE	UCS-8
UCS-4BE	UCS-16
UCS-4BE	UCS-16BE
UCS-4BE	UCS-16LE
UCS-4BE	UCS-32
UCS-4BE	UCS-32BE
UCS-4BE	UCS-32LE
UCS-4LE	646 (ISO 646)
UCS-4LE	8859-1 (ISO8859-1)
UCS-4LE	8859-2 (ISO8859-2)
UCS-4LE	8859-3 (ISO8859-3)
UCS-4LE	8859-4 (ISO8859-4)
UCS-4LE	8859-5 (ISO8859-5)
UCS-4LE	8859-6 (ISO8859-6)
UCS-4LE	8859-7 (ISO8859-7)
UCS-4LE	8859-8 (SO 8859-8)
UCS-4LE	8859-9 (ISO8859-9)
UCS-4LE	8859-10 (ISO8859-10)
UCS-4LE	8859-13 (ISO8859-13)
UCS-4LE	8859-14 (ISO8859-14)
UCS-4LE	8859-15 (ISO8859-15)
UCS-4LE	8859-16 (ISO8859-15)
UCS-4LE	KOI8-R

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UCS-4LE	KOI8-U
UCS-4LE	UCS-2
UCS-4LE	UCS-2BE
UCS-4LE	UCS-2LE
UCS-4LE	UTF-16
UCS-4LE	UTF-16BE
UCS-4LE	UTF-16LE
UCS-4LE	UTF-8
UTF-7	UCS-2
UTF-7	UCS-4
UTF-7	UCS-8
UTF-8	646 (ISO 646)
UTF-8	8859-1 (ISO8859-1)
UTF-8	8859-2 (ISO8859-2)
UTF-8	8859-3 (ISO8859-3)
UTF-8	8859-4 (ISO8859-4)
UTF-8	8859-5 (ISO8859-5)
UTF-8	8859-6 (ISO8859-6)
UTF-8	8859-7 (ISO8859-7)
UTF-8	8859-8 (ISO8859-8)
UTF-8	8859-9 (ISO8859-9)
UTF-8	8859-10 (ISO8859-10)
UTF-8	8859-11 (ISO8859-11)
UTF-8	8859-13 (ISO8859-13)
UTF-8	8859-14 (ISO8859-14)
UTF-8	8859-15 (ISO8859-15)
UTF-8	8859-16 (ISO8859-16)
UTF-8	eucJP

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UTF-8	gb2312
UTF-8	iso2022
UTF-8	ko_KR-euc
UTF-8	ko_KR-johap
UTF-8	ko_KR-johap92
UTF-8	ko_KR-iso2022–7
UTF-8	zh_TW-euc
UTF-8	zh_TW-iso2022–7
UTF-8	zh_TW-cp937
UTF-8	FujitsuJEF-ascii-code
UTF-8	FujitsuJEF-ascii-face
UTF-8	FujitsuJEF-kana-code
UTF-8	FujitsuJEF-kana-face
UTF-8	GBK
UTF-8	HitachiKEIS83
UTF-8	HitachiKEIS90
UTF-8	ISO-2022–JP
UTF-8	KOI8–R
UTF-8	KOI8-U
UTF-8	UTF-7
UTF-8	NECJIPS
UTF-8	PCK
UTF-8	UCS-2
UTF-8	UCS-2BE
UTF-8	UCS-2LE
UTF-8	UCS-4
UTF-8	UCS-4BE
UTF-8	UCS-4LE

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UTF-8	UTF-7
UTF-8	UTF-16
UTF-8	UTF-16BE
UTF-8	UCS-16LE
UTF-16	646 (ISO 646)
UTF-16	8859-1 (ISO8859-1)
UTF-16	8859-2 (ISO8859-2)
UTF-16	8859-3 (ISO8859-3)
UTF-16	8859-4 (ISO8859-4)
UTF-16	8859-5 (ISO8859-5)
UTF-16	8859-6 (ISO8859-6)
UTF-16	8859-7 (ISO8859-7)
UTF-16	8859-8 (ISO8859-8)
UTF-16	8859-9 (ISO8859-9)
UTF-16	8859-10 (ISO8859-10)
UTF-16	8859-13 (ISO8859-13)
UTF-16	8859-14 (ISO8859-14)
UTF-16	8859-15 (ISO8859-15)
UTF-16	8859-15 (ISO8859-15)
UTF-16	8859-16 (ISO8859-16)
UTF-16	KOI8-R
UTF-16	KOI8-U
UTF-16	UCS-4
UTF-16	UCS-4BE
UTF-16	UCS-4LE
UTF-16	UTF-8
UTF-16BE	646 (ISO 646)
UTF-16BE	8859-1 (ISO8859-1)

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UTF-16BE	8859-2 (ISO8859-2)
UTF-16BE	8859-3 (ISO8859-3)
UTF-16BE	8859-4 (ISO8859-4)
UTF-16BE	8859-5 (ISO8859-5)
UTF-16BE	8859-6 (ISO8859-6)
UTF-16BE	8859-7 (ISO8859-7)
UTF-16BE	8859-8 (ISO8859-8)
UTF-16BE	8859-9 (ISO8859-9)
UTF-16BE	8859-10(ISO8859-10)
UTF-16BE	8859-13 (ISO8859-13)
UTF-16BE	8859-14 (ISO8859-14)
UTF-16BE	8859-15 (ISO8859-15)
UTF-16BE	8859-16 (ISO8859-16)
UTF-16BE	KOI8-R
UTF-16BE	KOI8-U
UTF-16BE	UCS-4
UTF-16BE	UCS-4BE
UTF-16BE	UCS-4LE
UTF-16BE	UTF-8
UTF-16LE	646 (ISO 646)
UTF-16LE	8859-1 (ISO8859-1)
UTF-16LE	8859-2 (ISO8859-2)
UTF-16LE	8859-3 (ISO8859-3)
UTF-16LE	8859-4 (ISO8859-4)
UTF-16LE	8859-5 (ISO8859-5)
UTF-16LE	8859-6 (ISO8859-6)
UTF-16LE	8859-7 (ISO8859-7)
UTF-16LE	8859 -8 (ISO8859-8)

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UTF-16LE	8859-9 (ISO8859-9)
UTF-16LE	8859-10 (ISO8859-10)
UTF-16LE	8859-13 (ISO8859-13)
UTF-16LE	8859-14 (ISO8859-14)
UTF-16LE	8859-15 (ISO8859-15)
UTF-16LE	8859-16 (ISO8859-16)
UTF-16LE	KOI8-R
UTF-16LE	KOI8-U
UTF-16LE	UCS-4
UTF-16LE	UCS-4BE
UTF-16LE	UCS-4LE
UTF-16LE	UTF-8
UTF-32	UTF-8
UTF-32	UCS-2
UTF-32	UCS-2BE
UTF-32	UCS-2LE
UTF-32	UCS-4
UTF-32	UCS-4BE
UTF-32	UCS-4LE
UTF-32	UTF-16
UTF-32	UTF-16LE
UTF-32	UTF–32BE
UTF-32	646 (ISO 646)
UTF-32	ISO8859–1
UTF-32	ISO8859–2
UTF-32	ISO8859–3
UTF-32	ISO8859–4
UTF-32	ISO8859–5

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UTF-32	ISO8859–6
UTF-32	ISO8859–7
UTF-32	ISO8859–8
UTF-32	ISO8859–9
UTF-32	ISO8859–10
UTF-32	ISO8859–13
UTF-32	ISO8859–14
UTF-32	ISO8859–15
UTF-32	ISO8859–16
UTF-32	KOI8–R
UTF-32	KOI8–U
UTF-32BE	UTF-8
UTF-32BE	UCS-2
UTF-32BE	UCS-2BE
UTF-32BE	UCS-2LE
UTF-32BE	UCS-4
UTF-32BE	UCS-4BE
UTF-32BE	UCS-4LE
UTF-32BE	UTF-16
UTF–32BE	UTF-16BE
UTF-32 BE	UTF-16LE
UTF-32BE	646 (ISO 646)
UTF-32BE	ISO8859–1
UTF-32BE	ISO8859–2
UTF-32BE	ISO8859–3
UTF-32BE	ISO8859–4
UTF-32BE	ISO8859–5
UTF-32BE	ISO8859–6

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UTF-32BE	ISO8859–7
UTF-32BE	ISO8859–8
UTF-32BE	ISO8859–9
UTF-32BE	ISO8859–10
UTF-32BE	ISO8859–13
UTF-32BE	ISO8859–14
UTF-32BE	ISO8859–15
UTF-32BE	ISO8859–16
UTF-32BE	KOI8–R
UTF-32BE	KOI8–U
UTF-32LE	UTF-8
UTF-32LE	UCS-2
UTF-32LE	UCS-2BE
UTF-32LE	UCS-2LE
UTF-32LE	UCS-4
UTF-32LE	UCS-4BE
UTF-32LE	UCS-4LE
UTF32–LE	UTF-16
UTF32–LE	UTF-16BE
UTF-32LE	UTF-16LE
UTF-32LE	646 (ISO 646)
UTF-32LE	ISO8859–1
UTF-32LE	ISO8859–2
UTF-32LE	ISO8859–3
UTF-32LE	ISO8859–4
UTF-32LE	ISO8859–5
UTF-32LE	ISO8859–6
UTF-32LE	ISO8859–7

表 A–1 与 Unicode 相关的 iconv 可用代码转换模块　　(续下)

从代码（符号）	到代码（符号）
UTF-32LE	ISO8859–8
UTF-32LE	ISO8859–9
UTF-32LE	ISO8859–10
UTF-32LE	ISO8859–13
UTF-32LE	ISO8859–14
UTF-32LE	ISO8859–15
UTF-32LE	ISO8859–16
UTF-32LE	KOI8–R
UTF-32LE	KOI8–U

注意：UTF-EBCDIC 是一个新的 IBM 代码页名称。Solaris 9 环境还支持双向 UTF-8 <—> UTF-EBCDIC 转换。

下表列出了 Solaris 9 环境中可用的 Unicode 和 IBM/Microsoft EBCDIC 以及 PC iconv 代码转换模块。

表 A–2 与 Unicode 和 IBM/Microsoft EBCDIC 及 PC 代码页相关的 iconv 可用代码转换模块

从代码（符号）	到代码（符号）
UTF-8	IBM-037
UTF-8	IBM-273
UTF-8	IBM-277
UTF-8	IBM-278
UTF-8	IBM-280
UTF-8	IBM-284
UTF-8	IBM-285
UTF-8	IBM-297
UTF-8	IBM-420
UTF-8	IBM-424
UTF-8	IBM-500

表 A–2 与 Unicode 和 IBM/Microsoft EBCDIC 及 PC 代码页相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UTF-8	IBM-850
UTF-8	IBM-852
UTF-8	IBM-855
UTF-8	IBM-856
UTF-8	IBM-857
UTF-8	IBM-862
UTF-8	IBM-864
UTF-8	IBM-866
UTF-8	IBM-869
UTF-8	IBM-870
UTF-8	IBM-871
UTF-8	IBM-875
UTF-8	IBM-880
UTF-8	IBM-1025
UTF-8	IBM-1026
UTF-8	IBM-1112
UTF-8	IBM-1122
UTF-8	IBM-921
UTF-8	IBM-922
UTF-8	IBM-1046
UTF-8	IBM-1140
UTF-8	IBM-1141
UTF-8	IBM-1142
UTF-8	IBM-1143
UTF-8	IBM-1144
UTF-8	IBM-1145
UTF-8	IBM-1146

表 A–2 与 Unicode 和 IBM/Microsoft EBCDIC 及 PC 代码页相关的 iconv 可用代码转换模块 （续下）

从代码（符号）	到代码（符号）
UTF-8	IBM-1147
UTF-8	IBM-1148
UTF-8	IBM-1149
UTF-8	CP850
UTF-8	CP852
UTF-8	CP855
UTF-8	CP857
UTF-8	CP862
UTF-8	CP864
UTF-8	CP866
UTF-8	CP869
UTF-8	CP874
UTF-8	CP1250
UTF-8	CP1251
UTF-8	CP1252
UTF-8	CP1253
UTF-8	CP1254
UTF-8	CP1255
UTF-8	CP1256
UTF-8	CP1257
UTF-8	CP1258

下表列出了可用的 iconv 代码转换，可从 IBM 和 Microsoft EBCDIC/PC 代码页转换到 UTF-8。

表 A–3 可用的 iconv 代码转换 - IBM 和 Microsoft EBCDIC/PC 代码页到 UTF-8

UTF-EBCDIC	UTF-8
IBM-037	UTF-8
IBM-273	UTF-8

表 A–3 可用的 iconv 代码转换 - IBM 和 Microsoft EBCDIC/PC 代码页到 UTF-8 （续下）

UTF-EBCDIC	UTF-8
IBM-277	UTF-8
IBM-278	UTF-8
IBM-280	UTF-8
IBM-284	UTF-8
IBM-285	UTF-8
IBM-297	UTF-8
IBM-420	UTF-8
IBM-424	UTF-8
IBM-500	UTF-8
IBM-850	UTF-8
IBM-852	UTF-8
IBM-855	UTF-8
IBM-856	UTF-8
IBM-857	UTF-8
IBM-862	UTF-8
IBM-864	UTF-8
IBM-866	UTF-8
IBM-869	UTF-8
IBM-870	UTF-8
IBM-871	UTF-8
IBM-875	UTF-8
IBM-880	UTF-8
IBM-921	UTF-8
IBM-922	UTF-8
IBM-1025	UTF-8
IBM-1026	UTF-8
IBM-1046	UTF-8
IBM-1112	UTF-8

表 A–3 可用的 iconv 代码转换 - IBM 和 Microsoft EBCDIC/PC 代码页到 UTF-8 （续下）

UTF-EBCDIC	UTF-8
IBM-1122	UTF-8
IBM-1140	UTF-8
IBM-1141	UTF-8
IBM-1142	UTF-8
IBM-1143	UTF-8
IBM-1144	UTF-8
IBM-1145	UTF-8
IBM-1146	UTF-8
IBM-1147	UTF-8
IBM-1148	UTF-8
IBM-1149	UTF-8
CP850	UTF-8
CP852	UTF-8
CP855	UTF-8
CP857	UTF-8
CP862	UTF-8
CP864	UTF-8
CP866	UTF-8
CP869	UTF-8
CP874	UTF-8
CP1250	UTF-8
CP1251	UTF-8
CP1252	UTF-8
CP1253	UTF-8
CP1254	UTF-8
CP1255	UTF-8
CP1256	UTF-8
CP1257	UTF-8

表 A–3 可用的 iconv 代码转换 - IBM 和 Microsoft EBCDIC/PC 代码页到 UTF-8 （续下）

UTF-EBCDIC	UTF-8
CP1258	UTF-8

附录 B

Software CD 上的不完全语言环境软件包列表

下表列出了 Software CD 上的软件包和内容。

表 B–1 不完全语言环境列表

软件包名称	描述
JSatsvr	日语 Solaris 的日语输入系统 ATOK12 根文件。
JSatsvu	日语 Solaris 的日语输入系统 ATOK12 usr 文件。
JSatsvw	日语 Solaris 的日语输入系统 ATOK12 X11 支持文件。
NSCPcpcom	支持国际安全性的 Netscape Communicator 4.7 简体中文部分版本。
NSCPhpcom	支持国际安全性的 Netscape Communicator 4.7 的繁体中文部分版本。
NSCPjacom	支持国际安全性的 Netscape Communicator 4.7 日语（通用）本地化版本。
NSCPkpcom	支持国际安全性的 Netscape Communicator 4.78 朝鲜语部分版本。
SUNW5ttf	繁体中文 TrueType 字体软件包。
SUNWale	由中文、日语和朝鲜语语言环境共享的公用文件。它是运行亚洲语言环境所必需的软件包。
SUNWaled	中文、日语和朝鲜语语言环境共享的手册页。
SUNWalex	中文、日语和朝鲜语语言环境共享的公用文件。该软件包是运行亚洲语言环境（64 位）所必需的软件包。
SUNWauadt	澳大拉西亚 CDE 支持。
SUNWauaos	澳大拉西亚 OS 支持。
SUNWauaow	澳大拉西亚 Open Windows 支持。

表 B–1 不完全语言环境列表 （续下）

软件包名称	描述
SUNWauaox	澳大拉西亚 64 位 Open Windows 支持。
SUNWcamdt	中美洲 CDE 支持。
SUNWcamos	中美洲 OS 支持。
SUNWcamow	中美洲 Open Windows 支持。
SUNWcamox	中美洲 64 位 OS 支持。
SUNWcdt	CDE 桌面登录环境的简体中文本地化版本。
SUNWceudt	中欧 CDE 支持。
SUNWceuos	中欧 OS 支持。
SUNWceuow	中欧 Open Windows 支持。
SUNWceuox	中欧 64 位 OS 支持。
SUNWcleu	简体中文 (EUC) 语言环境的特定文件。它是运行简体中文 (EUC) 语言环境所必需的软件包。
SUNWcleux	简体中文语言环境的特定文件。它是运行简体中文语言环境（64 位）所必需的软件包。
SUNWcttf	简体中文 (EUC) True Type 字体。
SUNWcufnt	简体中文 (UTF-8) X Windows 平台所必需的字体。
SUNWcxplt	简体中文 X Windows 平台软件包。
SUNWdelu	
SUNWdespl	拼写检查引擎 - 德语词典。
SUNWeeudt	东欧 CDE 支持。
SUNWeeuos	东欧 OS 支持。
SUNWeeuow	东欧 Open Windows 支持。
SUNWeeuox	东欧 64 位 OS 支持。
SUNWeslu	实时升级本地化版本的西班牙语信息文件。
SUNWesspl	拼写检查引擎 - 西班牙语词典。
SUNWeu8df	美国英语/UTF-8 核心桌面文件。
SUNWeu8os	OS 环境用户文件的美国英语/UTF-8 本地化版本。
SUNWeu8ox	OS 环境用户文件（64 位）的美国英语/UTF-8 本地化版本。
SUNWeudba	CDE Base 的美国英语/UTF-8 本地化版本。

表 B–1 不完全语言环境列表 （续下）

软件包名称	描述
SUNWeudbd	CDE Dtbuilder 的美国英语/UTF-8 本地化版本。
SUNWeudda	CDE 桌面应用程序的美国英语/UTF-8 本地化版本。
SUNWeudhr	CDE 帮助运行时的美国英语/UTF-8 本地化版本。
SUNWeudhs	CDE 帮助运行时的美国英语/UTF-8 本地化版本。
SUNWeudis	CDE 图标的美国英语/UTF-8 本地化版本。
SUNWeudiv	桌面图像工具的美国英语/UTF-8 本地化版本。
SUNWeudlg	CDE 桌面登录环境的美国英语/UTF-8 本地化版本。
SUNWeudmg	桌面窗口管理器的美国英语/UTF-8 本地化版本。
SUNWeuezt	桌面电源组应用程序的美国英语/UTF-8 本地化版本。
SUNWeuhed	CDE 帮助开发者环境的美国英语/UTF-8 本地化版本。
SUNWeuluf	环境用户文件的美国英语/UTF-8 本地化版本。
SUNWeulux	环境用户文件（64 位）的美国英语/UTF-8 本地化版本。
SUNWeusru	Solaris 用户注册的美国英语/UTF-8 本地化版本。
SUNWfrlu	实时升级本地化版本的法语信息文件。
SUNWfrspl	拼写检查引擎 - 法语词典。
SUNWgttf	简体中文 (GBK) True Type 字体。
SUNWgxfnt	简体中文 (GBK) X Windows 平台所必需的字体。
SUNWgxplx	简体中文 (GBK) X Windows 平台软件包（64 位）。
SUNWhdt	CDE 桌面登录环境的繁体中文本地化版本。
SUNWhkdt	CDE 桌面登录环境的繁体中文（香港）本地化版本。
SUNWhkfnt	繁体中文 BIG5（香港）X Windows 平台所必需的字体软件包。
SUNWhkleu	繁体中文（香港）语言环境用户文件。
SUNWhklex	繁体中文（香港）语言环境用户文件（64 位）。
SUNWhkplt	繁体中文（香港）X Windows 平台软件包。
SUNWhkplx	繁体中文（香港）X Windows 平台软件包（64 位）。
SUNWhkttf	繁体中文香港补充字符集 True Type 字体软件包。
SUNWhleu	繁体中文语言环境的特定文件。它是运行繁体中文语言环境所必需的软件包。

表 B–1 不完全语言环境列表 （续下）

软件包名称	描述
SUNWhleux	繁体中文 (EUC) 语言环境的特定文件。它是运行繁体中文语言环境所必需的软件包（64 位）。
SUNWhttf	繁体中文 TrueType 字体软件包。
SUNWhufnt	繁体中文 (UTF-8) X Windows 平台所必需的字体。
SUNWhxfnt	繁体中文 X Windows 平台所必需的字体软件包。
SUNWhxplt	繁体中文 X Windows 平台软件包。
SUNWi1cs	X11 ISO8859-1 代码集支持。
SUNWi2cs	X11 ISO8859-2 代码集支持。
SUNWi2of	ISO-8859-2 字符集的 X11 字体（可选字体）。
SUNWi5cs	X11 ISO8859–5 代码集支持。
SUNWi7cs	X11 ISO8859–7 代码集支持。
SUNWi9cs	X11 ISO8859–9 代码集支持。
SUNWi13cs	X11 ISO8859-13 代码集支持。
SUNWi15cs	X11 ISO8859-15 代码集支持。
SUNWiiimr	因特网/内部网输入法框架 (Root)。
SUNWiiimu	因特网/内部网输入法框架 (Usr)。
SUNWindt	CDE 桌面登录环境的印度语本地化版本。
SUNWinfnt	印度语 (UTF-8) X Windows 平台所必需的字体。
SUNWinleu	印度语软件包包含印度语语言环境的特定文件。它是运行印度语语言环境所必需的软件包。
SUNWinlex	印度语软件包包含印度语语言环境的特定文件。它是运行印度语语言环境所必需的软件包（64 位）。
SUNWinplt	印度语 X Window 系统平台软件。
SUNWinttf	印度语 TrueType 字体。
SUNWitlu	实时升级本地化版本的意大利语信息文件。
SUNWitspl	拼写检查引擎 - 意大利语词典。
SUNWj3jmp	J2SDK RELEASE 手册页的日语本地化版本。
SUNWjedt	CDE 桌面登录环境的日语 (EUC) 本地化版本。
SUNWjexpx	X Window 系统平台软件（64 位）的日语 (EUC) 本地化版本。

表 B–1 不完全语言环境列表 （续下）

软件包名称	描述
SUNWjfpr	日语特征软件包 (JFP) 的流模块；它是运行 JFP 环境所必需的软件包。
SUNWjfpu	usr 的日语特征软件包 (JFP) 的特定文件；它是运行 JFP 环境所必需的软件包。
SUNWjfpux	usr 的日语特征软件包 (JFP) 的特定 64 位文件；它是运行 JFP 环境所必需的软件包。
SUNWjman	用于查看 SUNWjfpr 和 SUNWjfpu 的英语手册页的日语特征软件包手册页。
SUNWjulcf	日语 UTF-8 语言环境公用文件。
SUNWjxplt	X Window 系统平台软件的日语本地化版本。
SUNWkdt	CDE 桌面登录环境的朝鲜语本地化版本。
SUNWkleu	朝鲜语语言环境的特定文件。它是运行朝鲜语语言环境所必需的软件包。
SUNWkleux	朝鲜语 (EUC) 语言环境的特定文件。它是运行朝鲜语语言环境（64 位）所必需的软件包。
SUNWkttf	朝鲜语 True Type 字体。
SUNWkxfnt	朝鲜语 X Windows 平台所必需的字体
SUNWkxplt	朝鲜语 X Windows 平台软件包。
SUNWmeadt	中东 CDE 支持。
SUNWmeaos	中东 OS 支持。
SUNWmeaow	中东 OW 支持。
SUNWmeaox	中东 64 位 OS 支持。
SUNWnafos	北非 OS 支持。
SUNWnafow	北非 OW 支持。
SUNWnafox	北非 64 位 OS 支持。
SUNWneudt	北欧 CDE 支持。
SUNWneuos	北欧 OS 支持。
SUNWneuow	北欧 OW 支持。
SUNWneuox	北欧 64 位 OS 支持。
SUNWplow	不完全语言环境的 OpenWindows 启用。
SUNWplow1	补充的不完全语言环境的 OpenWindows 启用。

表 B–1 不完全语言环境列表 （续下）

软件包名称	描述
SUNWsamdt	南美 CDE 支持。
SUNWsamos	南美 OS 支持。
SUNWsamow	南美 OW 支持。
SUNWsamox	南美 64 位 OS 支持。
SUNWseudt	南欧 CDE 支持。
SUNWseuos	南欧 OS 支持。
SUNWseuow	南欧 Open Windows 支持。
SUNWseuox	南欧 64 位 OS 支持。
SUNWsvis	瑞典语安装软件本地化版本。
SUNWsvlu	实时升级本地化版本的瑞典语信息文件。
SUNWsvspl	拼写检查引擎 - 瑞典语词典。
SUNWtdbas	CDE Base 功能的泰语本地化版本。
SUNWtddst	CDE 桌面应用程序的泰语本地化版本。
SUNWtddte	CDE 桌面登录环境的泰语本地化版本。
SUNWtdft	CDE 字体的泰语本地化版本。
SUNWtdwm	CDE 桌面窗口管理器的泰语本地化版本。
SUNWtleu	泰语语言环境的特定文件。它是运行泰语语言环境所必需的软件包。
SUNWtleux	泰语语言环境的特定文件。它是运行泰语语言环境（64 位）所必需的软件包。
SUNWtxplt	泰语 X Windows 平台软件包。
SUNWweudt	西欧 CDE 支持。
SUNWweuos	西欧 OS 支持。
SUNWweuow	西欧 Open Windows 支持。
SUNWweuox	西欧 64 位 OS 支持。

附录 C

Language CD 上的完整语言环境软件包列表

下表列出了简体中文 Language 软件包及其内容。

表 C–1 简体中文

软件包名称	描述
IPLTcadcon	管理服务器控制台简体中文本地化版本。
IPLTcadman	管理服务器文档简体中文本地化版本。
IPLTcadmin	管理服务器简体中文本地化版本。
IPLTccons	控制台客户机基软件包简体中文本地化版本。
IPLTcdscon	目录服务器控制台简体中文本地化版本。
IPLTcdsman	目录服务器文档简体中文本地化版本。
IPLTcdsu	目录服务器简体中文本地化版本。
NSCPccom	支持国际安全性的 Netscape Communicator 4.7 简体中文本地化版本。
NSCPcucom	支持国际安全性的 Netscape Communicator 4.7 zh.UTF-8 本地化版本。
NSCPgcom	支持国际安全性的 Netscape Communicator 4.7 zh.GBK 本地化版本。
SUNWcacx	简体中文 AccessX 客户端程序。
SUNWcadis	用于 Admintool 和 GUI 安装的简体中文 (EUC) 本地化版本。
SUNWcadma	用来执行系统管理任务的软件的简体中文 (EUC) 本地化版本。为进行简体中文本地化，Admintool 既需要此软件包又需要 SUNWhadis 软件包。
SUNWcbcp	简体中文 (EUC) 语言环境二进制兼容性文件。
SUNWcdab	CDE 桌面应用程序生成器的简体中文 (EUC) 本地化版本。
SUNWcdbas	CDE 基本功能的简体中文 (EUC) 本地化版本。
SUNWcdcl	Solaris 无盘客户机管理应用程序的简体中文本地化版本。

表 C–1 简体中文 （续下）

软件包名称	描述
SUNWcddst	CDE 桌面应用程序的简体中文 (EUC) 本地化版本。
SUNWcddte	CDE 桌面登录环境的简体中文 (EUC) 本地化版本。
SUNWcdezt	桌面电源组应用程序的简体中文 (EUC) 本地化版本。
SUNWcdft	CDE 字体的简体中文 (EUC) 本地化版本。
SUNWcdhcm	DHCP 管理器的简体中文本地化版本。
SUNWcdhe	CDE 帮助运行时环境的简体中文 (EUC) 本地化版本。
SUNWcdhev	简体中文 (EUC) CDE 帮助卷。
SUNWcdhez	简体中文 (EUC)（通用）桌面电源组帮助卷。
SUNWcdicn	CDE 图标的简体中文 (EUC) 本地化版本。
SUNWcdim	CDE ImageTool 的简体中文 (EUC) 本地化版本。
SUNWcdwm	CDE 桌面窗口管理器的简体中文 (EUC) 本地化版本。
SUNWcepmw	电源管理 OW 公用程序的简体中文 (EUC) 本地化版本。
SUNWcfdl	Adobe Postcript TCP/IP 打印机的简体中文 Solaris 字体下载程序。
SUNWcj2p	Java Plug-In 1.2.2 的简体中文本地化版本。
SUNWcj2rt	Java 虚拟机和核心类库（简体中文增补）。
SUNWcjmfp	JMF 播放器的简体中文本地化版本
SUNWckcsr	简体中文 (EUC) KCMS 运行时环境。
SUNWcleex	简体中文语言环境的专用文件。 它是运行简体中文语言环境（64 位）扩展所必需的软件包。
SUNWcleue	简体中文 (EUC) 语言环境的专用文件。 它是运行简体中文 (EUC) 语言环境所必需的软件包。
SUNWclvma	Solaris 卷管理的简体中文本地化版本。
SUNWclvmg	Solaris 卷管理的简体中文本地化版本。
SUNWcmga	简体中文 Solaris 管理应用程序。
SUNWcorte	简体中文 (EUC) Open Look 工具箱运行时环境软件包。
SUNWcos	此包包含简体中文语言环境的专用文件。它是运行简体中文语言环境所必需的软件包。
SUNWcpdas	用于同步桌面应用程序和 Palm Pilot PDA 的工具的简体中文本地化版本。
SUNWcreg	用于 Solaris 用户注册的简体中文 (EUC) 本地化版本。
SUNWcrmui	简体中文资源管理用户界面组件。

表 C–1 简体中文 （续下）

软件包名称	描述
SUNWcsadl	Solstice Admintool 启动器及其关联库的简体中文 (EUC) 本地化版本。
SUNWcscgu	简体中文可本地化的 Solaris 智能卡管理 - 图形用户界面组件。
SUNWcsfw	用于 SFW 统一的简体中文可本地化信息文件。
SUNWcsmc	简体中文 Solaris 管理控制台 2.0。
SUNWctltk	简体中文 (EUC) ToolTalk 运行时软件包。
SUNWcttfe	简体中文 (EUC) TrueType 字体。
SUNWcudc	Solaris CDE 环境的用户定义的字符工具的简体中文 (EUC) 本地化版本。
SUNWcwbc	Solaris WBEM 服务的简体中文本地化版本。
SUNWcwbcp	简体中文 (EUC) OpenWindows 二进制兼容性软件包。
SUNWcwdev	Solaris WBEM 服务的简体中文本地化版本。
SUNWcwsr2	Solaris 产品注册表的简体中文本地化版本。
SUNWcwsrv	Solaris 产品注册表查看器的简体中文本地化版本。
SUNWcxe	简体中文 (EUC) X Windows 平台软件包。
SUNWcxfnt	简体中文 (EUC) X Windows 平台必需的字体。
SUNWcxman	简体中文 (EUC) X Windows 联机用户手册页软件包。
SUNWgttfe	简体中文 (GBK) True Type 字体。

下表列出了法语语言软件包及其内容。

表 C–2 法语

软件包名称	描述
NSCPfrcdo	支持美国安全性的 Netscape Communicator 4.7 法语本地化版本。
NSCPfrcom	支持国际安全性的 Netscape Communicator 4.7 法语本地化版本。
SUNWfbcp	法语 OS 二进制兼容性软件包。
SUNWfdcl	无盘客户机管理应用程序的法语本地化版本。
SUNWfdhcm	DHCP 管理器的法语本地化版本。
SUNWffdl	字体下载程序的可本地化字符串。
SUNWfj2rt	Java 虚拟机和核心类库（法语增补）。
SUNWfjmfp	用于播放音频和视频文件的可本地化 JMF 播放器。

表 C–2 法语　（续下）

软件包名称	描述
SUNWflvma	Solaris 卷管理 API 的法语本地化版本。
SUNWflvmg	Solaris 卷管理应用程序的法语本地化版本。
SUNWfmgp	Solaris 管理应用程序法语本地化版本。
SUNWforte	法语 OPEN LOOK (R) 工具箱运行时环境。
SUNWfpdas	用于同步桌面应用程序和 Palm Pilot PDA 的法语工具。
SUNWfrbas	运行 CDE 应用程序的基本本地化版本法语 CDE 功能。
SUNWfrdis	用于 Admintool 和 GUI 安装的法语本地化版本。
SUNWfrdma	用于执行系统管理任务的软件的法语本地化版本。
SUNWfrdst	CDE 桌面应用程序。
SUNWfrdte	CDE 桌面环境。
SUNWfrhe	CDE 帮助本地化版本的法语运行时环境。
SUNWfrhed	CDE 本地化版本的法语帮助开发者环境。
SUNWfrhev	CDE 帮助卷。
SUNWfrhez	桌面电源组帮助卷的法语本地化版本。
SUNWfrim	CDE 桌面应用程序。
SUNWfrj2p	Java Plug-in 1.2.2 的法语本地化版本。
SUNWfrmui	Solaris 的法语资源管理。
SUNWfros	用于 OS 联网统一的可本地化信息文件。
SUNWfrpmw	电源管理 Open Windows 公用程序的法语 (EUC) 本地化版本。
SUNWfrreg	在桌面登录时为进行用户注册的 Solaris 用户注册提示。
SUNWfrsmc	Solaris 管理控制台的法语本地化版本。
SUNWfrwbc	Solaris WBEM 服务的法语本地化版本。
SUNWfrwm	法语 CDE 桌面窗口管理器信息。
SUNWfrws2	Solaris 产品注册的法语本地化版本。
SUNWfrwsv	Solaris 产品注册查看器的法语本地化版本。
SUNWfsadl	Solstice Admintool 启动器及其关联的库的法语本地化版本。
SUNWfscgu	Solaris 智能卡管理 - 图形用户界面组件的法语本地化版本。
SUNWfsfw	用于 SFW 统一的法语可本地化信息文件。

表 C–2 法语 （续下）

软件包名称	描述
SUNWftltk	法语 ToolTalk 二进制和共享库。
SUNWfwacx	法语 OPEN LOOK (R) AccessX。
SUNWfwbcp	法语 OpenWindows 二进制兼容性软件包。
SUNWfwdev	法语 Sun WBEM SDK 资源。
SUNWfxplt	法语 X Windows 平台软件。

下表列出了德语语言软件包及其内容。

表 C–3 德语

软件包名称	描述
NSCPdecom	支持国际安全性的 Netscape Communicator 4.7 德语本地化版本。
SUNWdbcp	德语 OS 二进制兼容性软件包。
SUNWddcl	Solaris 无盘客户机管理应用程序的德语本地化版本。
SUNWddhcm	DHCP 管理器的德语本地化版本。
SUNWdebas	运行 CDE 应用程序的基本本地化版本德语 CDE 功能。
SUNWdedis	用于 Admintool 和 GUI 安装的德语本地化版本。
SUNWdedma	用于执行系统管理任务的软件的德语本地化版本。
SUNWdedst	CDE 桌面应用程序。
SUNWdedte	CDE 桌面登录环境。
SUNWdehe	CDE 帮助本地化版本德语运行时环境。
SUNWdehed	CDE 本地化版本的德语帮助开发者环境。
SUNWdehev	CDE 帮助卷。
SUNWdehez	桌面电源组帮助卷的德语本地化版本。
SUNWdeim	CDE 桌面应用程序。
SUNWdej2p	Java Plug-in 1.2.2 的德语本地化版本。
SUNWdeos	用于 OS 联网统一的可本地化信息文件。
SUNWdepmw	电源管理 OW 公用程序的德语 (EUC) 本地化版本。
SUNWdereg	在桌面登录时为进行用户注册的 Solaris 用户注册提示。
SUNWdesmc	Solaris 管理控制台的德语本地化版本。

表 C–3 德语　（续下）

软件包名称	描述
SUNWdewbc	Solaris WBEM 服务的德语本地化版本。
SUNWdewm	德语 CDE 桌面窗口管理器信息。
SUNWdews2	Solaris 产品注册的德语本地化版本。
SUNWdewsv	Solaris 产品注册查看器的德语本地化版本。
SUNWdfdl	字体下载程序的可本地化字符串。
SUNWdj2rt	Java 虚拟机和核心类库（德语增补）
SUNWdjmfp	用于播放音频和视频文件的可本地化 JMF 播放器。
SUNWdlvma	Solaris 卷管理 API 的德语本地化版本。
SUNWdlvmg	Solaris 卷管理应用程序的德语本地化版本。
SUNWdmgp	Solaris 管理应用程序的德语本地化版本。
SUNWdorte	德语 OPEN LOOK® 工具箱运行时环境。
SUNWdpdas	用于同步桌面应用程序和 Palm Pilot PDA 的德语工具。
SUNWdrmui	Solaris 的德语资源管理。
SUNWdsadl	Solstice Admintool 启动器及其关联的库的德语本地化版本。
SUNWdscgu	可本地化的 Solaris 智能卡管理 - 图形用户界面组件。
SUNWdsfw	用于 SFW 统一的德语可本地化信息文件。
SUNWdtltk	德语 ToolTalk 二进制和共享库。
SUNWdwacx	德语 OPEN LOOK® AccessX。
SUNWdwbcp	德语 OpenWindows 二进制兼容性软件包。
SUNWdwdev	德语 Sun WBEM SDK 资源。
SUNWdxplt	德语 X Windows 平台软件。

下表列出了意大利语语言软件包及其内容。

表 C–4 意大利语

软件包名称	描述
NSCPitcom	支持国际安全性的 Netscape Communicator 4.78 意大利语本地化版本。
SUNWibcp	意大利语 OS 二进制兼容性软件包。
SUNWidcl	无盘客户机管理应用程序的意大利语本地化版本。

表 C–4 意大利语 （续下）

软件包名称	描述
SUNWidhcm	DHCP 管理器的意大利语本地化版本。
SUNWifdl	字体下载程序的可本地化字符串。
SUNWij2rt	Java 虚拟机和核心类库（意大利语增补）。
SUNWijmfp	用于播放音频和视频文件的可本地化 JMF 播放器。
SUNWilvma	Solaris 卷管理 API 的意大利语本地化版本。
SUNWilvmg	Solaris 卷管理应用程序的意大利语本地化版本。
SUNWimgp	Solaris 管理应用程序的意大利语本地化版本。
SUNWiorte	意大利语 OPEN LOOK (R) 工具箱运行时环境。
SUNWipdas	用于同步桌面应用程序和 Palm Pilot PDA 的意大利语工具。
SUNWirmui	Solaris 的意大利语资源管理。
SUNWisadl	Solstice Admintool 启动器及其关联的库的意大利语本地化版本。
SUNWiscgu	可本地化的 Solaris 智能卡管理 - 图形用户界面组件。
SUNWisfw	用于 SFW 统一的意大利语可本地化信息文件。
SUNWitbas	运行 CDE 应用程序的基本本地化版本意大利语 CDE 功能。
SUNWitdis	用于 Admintool 和 GUI 安装的意大利语本地化版本。
SUNWitdma	用于执行系统管理任务的软件的意大利语本地化版本。
SUNWitdst	CDE 意大利语桌面应用程序信息。
SUNWitdte	CDE 意大利语桌面登录环境。
SUNWithe	CDE 帮助本地化版本意大利语运行时环境。
SUNWithed	CDE 本地化版本的意大利语帮助开发者环境。
SUNWithev	CDE 帮助卷。
SUNWithez	CDE 帮助卷的意大利语本地化版本。
SUNWitim	CDE 意大利语桌面图像编辑器。
SUNWitj2p	Java Plug-in 1.2.2 的意大利语本地化版本。
SUNWitltk	意大利语 ToolTalk 二进制和共享文件。
SUNWitos	用于 OS 联网统一的可本地化信息文件。
SUNWitpmw	电源管理 OW 公用程序的意大利语 (EUC) 本地化版本。
SUNWitreg	在桌面登录时为进行用户注册的 Solaris 用户注册提示。

表 C–4 意大利语 （续下）

软件包名称	描述
SUNWitsmc	Solaris 管理控制台的意大利语本地化版本。
SUNWitwbc	Solaris WBEM 服务的意大利语本地化版本。
SUNWitwm	意大利语 CDE 桌面窗口管理器信息。
SUNWitws2	Solaris 产品注册的意大利语版本。
SUNWitwsv	Solaris 产品注册查看器的意大利语版本。
SUNWiwacx	意大利语 OPEN LOOK (R) AccessX。
SUNWiwbcp	意大利语 OpenWindows 二进制兼容性软件包。
SUNWiwdev	意大利语 Sun WBEM SDK 资源。
SUNWixplt	意大利语 X Windows 平台软件。

下表列出了日语语言软件包及其内容。

表 C–5 日语

软件包名称	描述
LTjadcon	管理服务器控制台客户机软件包日语本地化版本。
IPLTjadman	管理服务器文档日语本地化版本。
IPLTjadmin	管理服务器日语本地化版本。
IPLTjcons	控制台客户机基软件包日语本地化版本。
IPLTjdscon	目录服务器控制台日语本地化版本。
IPLTjdsman	目录服务器文档日语本地化版本。
IPLTjdsu	目录服务器日语本地化版本。
NSCPjecom	支持国际安全性的 Netscape Communicator 4.7 日语 (EUC) 本地化版本。
NSCPjpcom	支持国际安全性的 Netscape Communicator 4.7 日语 (PCK) 本地化版本。
NSCPjucom	支持国际安全性的 Netscape Communicator 4.7 日语 (UTF-8) 本地化版本。
SUNWjadcl	Solaris 无盘客户机管理应用程序的日语本地化版本。
SUNWjadis	用于 Admintool 和 GUI 安装的日语 (EUC) 本地化版本。
SUNWjadma	用于执行系统管理任务的软件的日语 (EUC) 本地化版本。 为进行日语 (EUC) 本地化，Admintool 既需要此软件包又需要 SUNWjadis 软件包。
SUNWjaj2p	Java Plug-in 1.2.2 的日语本地化版本。

表 C–5 日语　（续下）

软件包名称	描述
SUNWjbcp	日语 (EUC) 公用程序，包括 libc 和语言环境数据，用以向 SunOS 4.x 应用程序提供一个二进制兼容的执行环境。
SUNWjcs3f	用于打印的日语 JIS X0212 Type1 字体。
SUNWjdab	CDE 桌面应用程序生成器的日语（通用）本地化版本。
SUNWjdbas	CDE 应用程序基本运行时环境的日语（通用）本地化版本。
SUNWjddst	CDE 桌面应用程序的日语 (EUC) 本地化版本。
SUNWjddte	Solaris 桌面登录环境的日语 (EUC) 本地化版本
SUNWjdhcm	DHCP 管理器的日语本地化版本
SUNWjdhe	CDE 帮助运行时环境的日语 (EUC) 本地化版本
SUNWjdhed	CDE 帮助开发者环境的日语 (EUC) 本地化版本
SUNWjdhev	CDE 帮助卷的日语（通用）本地化版本
SUNWjdhez	桌面电源组帮助卷的日语（通用）本地化版本
SUNWjdim	Solaris CDE 图像查看器的日语 (EUC) 本地化版本。
SUNWjdma	CDE 手册页的日语本地化版本。
SUNWjdoc	日语文档工具。
SUNWjdwm	CDE 桌面窗口管理器的日语 (EUC) 本地化版本。
SUNWject	UTF-8 代码转换工具的日语 (EUC) 本地化版本。
SUNWjedev	日语 (EUC) 开发环境软件包的专用文件。
SUNWjeezt	桌面电源组应用程序的日语 (EUC) 本地化版本。
SUNWjej2m	日语 (EUC) 手册页。
SUNWjeman	用于查看 SUNWjfpr 和 SUNWjfpu 的日语 (EUC) 手册页以及查看 SUNWman 和 SUNWaled 的日语手册页的日语特征软件包手册页。
SUNWjepmm	日语 (EUC) 电源管理 OW 公用程序手册页。
SUNWjepmw	电源管理 OW 公用程序的日语 (EUC) 本地化版本。
SUNWjeudc	Solaris CDE 环境的用户定义字符工具日语 (EUC) 本地化版本。
SUNWjfdl	Adobe PostScript 打印机的 Solaris 桌面字体下载程序日语本地化版本。
SUNWjfpre	日语特征软件包 (JFP) 的流模块。 它是一个用于运行 JFP 环境的可扩展软件包。
SUNWjfpue	面向用户的日语特征软件包 (JFP) 的专用文件。它是一个用于运行 JFP 环境的可扩展软件包。
SUNWjfxmn	X Window 系统的日语特征的英文手册页。

表 C–5 日语 （续下）

软件包名称	描述
SUNWjj2rt	日语 Java 虚拟机和核心类库。
SUNWjjmfp	JMF 播放器的日语本地化版本。
SUNWjkcsr	Kodak 颜色管理系统运行时的日语 (EUC) 本地化版本。
SUNWjlvma	Solaris 卷管理 API 的日语本地化版本。
SUNWjlvmg	Solaris 卷管理应用程序的日语本地化版本。
SUNWjmane	用于查看 SUNWjfpre 和 SUNWjfpue 的英文手册页的日语特征软件包手册页（扩展）。
SUNWjmfrn	Motif 1.2.3 运行时工具箱的日语 (EUC) 本地化版本。
SUNWjmga	日语 Solaris 管理应用程序。
SUNWjorte	OPEN LOOK 工具箱运行时环境的日语 (EUC) 本地化版本。
SUNWjos	日语 ON 信息文件。
SUNWjpdas	用于同步桌面应用程序和 Palm Pilot PDA 的工具的日语本地化版本。
SUNWjpj2m	日语 (PCK) 手册页。
SUNWjreg	用于 Solaris 用户注册的日语本地化版本。
SUNWjrmui	Solaris 的日语资源管理。
SUNWjsadl	Solstice Admintool 启动程序和相关库的日语 (EUC) 本地化版本。
SUNWjscag	Solaris 智能卡管理 - 图形用户界面组件的日语本地化版本。
SUNWjsfw	用于 SFW 统一的日语可本地化信息文件。
SUNWjsmc	日语 Solaris 管理控制台 2.0。
SUNWjtlmn	供 ToolTalk 程序员、OpenWindows 用户和通用桌面环境 (CDE) 用户使用的日语 (EUC) ToolTalk 手册页。
SUNWjtltk	通用桌面环境 (CDE)、OpenWindows 和所有 ToolTalk 客户机所需的 ToolTalk 二进制和共享库的日语本地化版本。
SUNWjuj2m	日语 (UTF-8) 手册页。
SUNWjwacx	AccessX 客户机程序的日语 (EUC) 本地化版本。
SUNWjwbc	Solaris WBEM 服务的日语本地化版本。
SUNWjwbcp	OpenWindows 二进制兼容性的支持文件、程序和库的日语 (EUC) 本地化版本。
SUNWjwbd	Sun WBEM SDK 资源的日语本地化版本。
SUNWjwncr	日语输入系统 - Wnn6 客户机 (Root)。
SUNWjwncu	日语输入系统 - Wnn6 客户机 (Usr)。

表 C–5 日语 （续下）

软件包名称	描述
SUNWjwncx	日语输入系统 - Wnn6 客户机 X Window 系统。
SUNWjwndt	日语输入系统 - 用于 CDE 的 Wnn6 客户机。
SUNWjwnsr	日语输入系统 - Wnn6 服务器 (Root)。
SUNWjwnsu	日语输入系统 - Wnn6 服务器 (Usr)。
SUNWjws2	Solaris 产品注册表的日语本地化版本。
SUNWjwsv	Solaris 产品注册查看器的日语本地化版本。
SUNWjxfa	Solaris 平台的字体管理应用程序的日语（通用）本地化版本。
SUNWjxfnt	日语 X Window 系统字体（必需的字体）- gothic 黑体字体和 TrueType 映射文件。
SUNWjxim	日语 X Window 系统 X 输入法服务器软件包。
SUNWjxoft	Sun Minchou 位图字体。
SUNWjxpmn	日语 (EUC) X Window 系统联机程序员手册页。
SUNWjxumn	日语 (EUC) X Window 系统联机用户手册页。

下表列出了朝鲜语语言软件包及其内容。

表 C–6 朝鲜语

软件包名称	描述
NSCPkocom	支持国际安全性的 Netscape Communicator 4.7 朝鲜语本地化版本。
NSCPkucom	支持国际安全性的 Netscape Communicator 4.78 ko.UTF-8 本地化版本。
SUNWkacx	朝鲜语 AccessX 客户机程序。
SUNWkadis	用于 Admintool 和 GUI 安装的朝鲜语本地化版本。
SUNWkadma	用来执行系统管理任务的软件的朝鲜语本地化版本。为进行朝鲜语本地化，Admintool 既需要此软件包又需要 SUNWkadis 软件包。
SUNWkbcp	该软件包包含韩语环境二进制兼容性文件。
SUNWkcoft	朝鲜语/朝鲜语 UTF-8 通用可选字体软件包。
SUNWkdab	CDE 桌面应用程序生成器的朝鲜语本地化版本。
SUNWkdbas	CDE 基本功能的朝鲜语本地化版本。
SUNWkdcl	Solaris 无盘客户机管理应用程序的朝鲜语本地化版本。
SUNWkdcst	朝鲜语的本地化工具软件包。
SUNWkddst	CDE 桌面应用程序的朝鲜语本地化版本。

表 C–6 朝鲜语 （续下）

软件包名称	描述
SUNWkddte	CDE 桌面登录环境的朝鲜语本地化版本。
SUNWkdezt	桌面电源组应用程序的朝鲜语 (EUC) 本地化版本。
SUNWkdft	通用桌面环境朝鲜语本地化版本 CDE 的字体。
SUNWkdhcm	DHCP 管理器的日语本地化版本。
SUNWkdhe	CDE 帮助运行时环境的朝鲜语本地化版本。
SUNWkdhev	朝鲜语 CDE 帮助卷。
SUNWkdhez	桌面电源组帮助卷的朝鲜语（通用）本地化版本。
SUNWkdicn	CDE 图标的朝鲜语本地化版本。
SUNWkdim	CDE Imagetool 的朝鲜语本地化版本。
SUNWkdwm	CDE 桌面窗口管理器的朝鲜语本地化版本。
SUNWkepmw	电源管理 OW 公用程序的朝鲜语 (EUC) 本地化版本。
SUNWkfdl	Adobe Postscript (tm) TCP/IP 打印机的朝鲜语 Solaris 字体下载程序。
SUNWkj2rt	Java 虚拟机和核心类库（朝鲜语增补）。
SUNWkjmfp	JMF 播放器的朝鲜语本地化版本。
SUNWkkcsr	朝鲜语 (EUC) KCMS 运行时环境。
SUNWkleex	朝鲜语语言环境的特定文件。它是运行朝鲜语语言环境（64 位）所必需的软件包。
SUNWkler	该软件包包含了朝鲜语语言环境的流模块。它是运行朝鲜语语言环境所必需的软件包。
SUNWkleue	该软件包包含了朝鲜语语言环境的专用文件。它是运行朝鲜语语言环境所必需的软件包。
SUNWklvma	Solaris 卷管理 API 的朝鲜语本地化版本。
SUNWklvmg	Solaris 卷管理应用程序的朝鲜语本地化版本。
SUNWkmga	朝鲜语 Solaris 管理应用程序。
SUNWkoj2p	Java Plug-in 1.2.2 的朝鲜语本地化版本。
SUNWkorte	朝鲜语 OPENLOOK 工具箱运行时环境软件包。
SUNWkos	该软件包包含了朝鲜语语言环境的专用文件。它是运行朝鲜语语言环境所必需的软件包。
SUNWkpdas	用于同步桌面应用程序和 Palm Pilot PDA 的工具的朝鲜语本地化版本。
SUNWkreg	用于 Solaris 用户注册的朝鲜语本地化版本。

表 C–6 朝鲜语 （续下）

软件包名称	描述
SUNWkrmui	朝鲜语资源管理用户界面组件。
SUNWksadl	Solstice Admintool 启动程序和相关库的朝鲜语本地化版本。
SUNWkscgu	朝鲜语可本地化的 Solaris 智能卡管理 - 图形用户界面组件。
SUNWksfw	用于 SFW 统一的朝鲜语可本地化信息文件。
SUNWksmc	朝鲜语 Solaris 管理控制台 2.0。
SUNWktltk	朝鲜语 ToolTalk 运行时软件包。
SUNWkttfe	朝鲜语 True Type 字体扩展。
SUNWkudc	Solaris CDE 环境的用户定义字符工具朝鲜语 (EUC) 本地化版本。
SUNWkuxft	朝鲜语 UTF-8 X Windows 平台必需的字体。
SUNWkwbc	Solaris WBEM 服务的朝鲜语本地化版本。
SUNWkwbcp	朝鲜语 OpenWindows 二进制兼容性软件包。
SUNWkwdev	Solaris WBEM 服务的朝鲜语本地化版本。
SUNWkwsr	朝鲜语 prodreg 2.0 可本地化的文本资源。
SUNWkwsr2	Solaris 产品注册的朝鲜语本地化版本。
SUNWkwsrv	Solaris 产品注册表查看器的朝鲜语本地化版本。
SUNWkxe	朝鲜语 X Windows 平台软件包。
SUNWkxfte	朝鲜语 X Windows 平台必需的字体
SUNWkxman	朝鲜语 X Windows 联机用户手册页软件包

下表列出了西班牙语语言软件包及其内容。

表 C–7 西班牙语

软件包名称	描述
NSCPescom	支持国际安全性的 Netscape Communicator 4.7 西班牙语本地化版本。
SUNWedcl	无盘客户机管理应用程序的西班牙语本地化版本。
SUNWedhcm	DHCP 管理器的西班牙语本地化版本。
SUNWefdl	字体下载程序的可本地化字符串。
SUNWej2rt	Java 虚拟机和核心类库（西班牙语增补）。
SUNWejmfp	用于播放音频和视频文件的可本地化 JMF 播放器。

表 C–7 西班牙语 （续下）

软件包名称	描述
SUNWelvma	Solaris 卷管理 API 的西班牙语本地化版本。
SUNWelvmg	Solaris 卷管理应用程序的西班牙语本地化版本。
SUNWemgp	Solaris 管理应用程序西班牙语本地化版本。
SUNWeorte	西班牙语 OPEN LOOK (R) 工具箱运行时环境。
SUNWepdas	用于同步桌面应用程序和 Palm Pilot PDA 的工具的西班牙语本地化版本。
SUNWermui	Solaris 的西班牙语资源管理。
SUNWesadl	Solstice Admintool 启动程序和相关库的西班牙语本地化版本。
SUNWesbas	运行 CDE 应用程序的基本本地化版本法语 CDE 功能。
SUNWescgu	可本地化的 Solaris 智能卡管理 - 图形用户界面组件。
SUNWesdis	用于 Admintool 和 GUI 安装的西班牙语本地化版本。
SUNWesdma	用来执行系统管理任务的软件的西班牙语本地化版本。
SUNWesdst	CDE 桌面应用程序。
SUNWesdte	CDE 桌面登录环境。
SUNWesfw	用于 SFW 统一的西班牙语可本地化信息文件。
SUNWeshe	CDE 帮助本地化版本的西班牙语运行时环境。
SUNWeshed	CDE 本地化版本的西班牙语帮助开发者环境。
SUNWeshev	CDE 帮助卷。
SUNWeshez	桌面电源组帮助卷的西班牙语本地化版本。
SUNWesim	CDE 桌面应用程序。
SUNWesj2p	Java Plug-in 1.2.2 的西班牙语本地化版本。
SUNWesos	用于 OS 联网统一的可本地化信息文件。
SUNWespmw	电源管理 OW 公用程序的西班牙语 (EUC) 本地化版本。
SUNWesreg	在桌面登录时为进行用户注册的 Solaris 用户注册提示。
SUNWessmc	Solaris 管理控制台西班牙语本地化版本。
SUNWeswbc	Solaris WBEM 服务的西班牙语本地化版本。
SUNWeswm	西班牙语 CDE 桌面窗口管理器信息。
SUNWesws2	西班牙语本地化版本的 Solaris 产品注册表。
SUNWeswsv	Solaris 产品注册表查看器的西班牙语本地化版本。

表 C–7 西班牙语 （续下）

软件包名称	描述
SUNWetltk	西班牙语 ToolTalk 二进制和共享库。
SUNWewacx	西班牙语 OPEN LOOK (R) AccessX。
SUNWewdev	西班牙语 Sun WBEM SDK 资源。
SUNWexplt	西班牙语 X Windows 平台软件。

下表列出了瑞典语语言软件包及其内容。

表 C–8 瑞典语

软件包名称	描述
NSCPsvcom	支持国际安全性的 Netscape Communicator 4.7 瑞典语本地化版本。
SUNWsdcl	无盘客户机管理应用程序的瑞典语本地化版本。
SUNWsdhcm	DHCP 管理器的瑞典语本地化版本。
SUNWsfdl	字体下载程序的可本地化字符串。
SUNWsj2rt	Java 虚拟机和核心类库（瑞典语增补）。
SUNWsjmfp	用于播放音频和视频文件的可本地化 JMF 播放器。
SUNWslvma	Solaris 卷管理 API 的瑞典语本地化版本。
SUNWslvmg	Solaris 卷管理应用程序的瑞典语本地化版本。
SUNWsmgp	Solaris 管理应用程序瑞典语本地化版本。
SUNWsorte	瑞典语 OPEN LOOK (R) 工具箱运行时环境。
SUNWspdas	用于同步桌面应用程序和 Palm Pilot PDA 的瑞典语工具。
SUNWsrmui	Solaris 的瑞典语资源管理。
SUNWssadl	Solstice Admintool 启动程序和相关库的瑞典语本地化版本。
SUNWsscgu	可本地化的 Solaris 智能卡管理 - 图形用户界面组件。
SUNWssfw	用于 SFW 统一的瑞典语可本地化信息文件。
SUNWstltk	瑞典语 ToolTalk 二进制和共享库。
SUNWsvbas	基本瑞典语 CDE 功能信息。
SUNWsvdis	用于 Admintool 和 GUI 安装的瑞典语本地化版本。
SUNWsvdma	用来执行系统管理任务的软件的瑞典语本地化版本。
SUNWsvdst	瑞典语 CDE 桌面应用程序信息。

表 C–8 瑞典语 （续下）

软件包名称	描述
SUNWsvdte	瑞典语 CDE 桌面登录环境信息。
SUNWsvhe	瑞典语 CDE 帮助运行时环境。
SUNWsvhed	瑞典语 CDE 帮助开发者环境信息。
SUNWsvhev	CDE 帮助卷。
SUNWsvhez	桌面电源组帮助卷的瑞典语本地化版本。
SUNWsvim	瑞典语 CDE 图像编辑器信息。
SUNWsvj2p	Java Plug-in 1.2.2 的瑞典语本地化版本。
SUNWsvos	用于 OS 联网统一的可本地化信息文件。
SUNWsvpmw	电源管理 Open Windows 的瑞典语 (EUC) 本地化版本。公用程序
SUNWsvreg	在桌面登录时为进行用户注册的 Solaris 用户注册提示。
SUNWsvsmc	Solaris 管理控制台瑞典语本地化版本。
SUNWsvwbc	Solaris WBEM 服务的瑞典语本地化版本。
SUNWsvwm	瑞典语 CDE 桌面窗口管理器信息。
SUNWsvws2	瑞典语本地化版本 Solaris 产品注册表。
SUNWsvwsv	Solaris 产品注册表查看器的瑞典语本地化版本。
SUNWsvwsv	Solaris 产品注册表查看器的瑞典语本地化版本。
SUNWswacx	瑞典语 Open Look AccessX。
SUNWsxplt	瑞典语 X Windows 平台软件。
SUNWvbcp	瑞典语 OS 二进制兼容性软件包。

下表列出了繁体中文语言软件包及其内容。

表 C–9 繁体中文

软件包名称	描述
NSCP5com	支持国际安全性的 Netscape Communicator 4.7 zh_TW.BIG5 本地化版本。
NSCPhcom	支持国际安全性的 Netscape Communicator 4.7 繁体中文本地化版本。
NSCPhucom	支持国际安全性的 Netscape Communicator 4.7 zh_TW.UTF-8 本地化版本。
SUNW5ttfe	繁体中文 True Type 字体软件包扩展。
SUNW5xfnt	繁体中文 BIG5 X Windows 平台必需的字体软件包。

表 C–9 繁体中文 （续下）

软件包名称	描述
SUNWhacx	繁体中文 AccessX 客户机程序。
SUNWhadis	用于 Admintool 和 GUI 安装的繁体中文 (EUC) 本地化版本。
SUNWhadma	用来执行系统管理任务的软件的繁体中文 (EUC)本地化版本。为进行繁体中文 (EUC) 本地化，Admintool 既需要此软件包又需要 SUNWhadis 软件包。
SUNWhbcp	该软件包包含繁体中文语言环境的二进制兼容性文件。
SUNWhdab	CDE 桌面应用程序生成器的繁体中文本地化版本。
SUNWhdbas	CDE 基本功能的繁体中文本地化版本。
SUNWhdcl	Solaris 无盘客户机管理应用程序的繁体中文本地化版本。
SUNWhddst	CDE 桌面应用程序的繁体中文本地化版本。
SUNWhddte	CDE 桌面登录环境的繁体中文本地化版本。
SUNWhdezt	桌面电源组应用程序的繁体中文 (EUC) 本地化版本。
SUNWhdft	CDE 字体的繁体中文本地化版本。
SUNWhdhcm	DHCP 管理器的繁体中文本地化版本。
SUNWhdhe	CDE 帮助运行时环境的繁体中文本地化版本。
SUNWhdhev	繁体中文 CDE 帮助卷
SUNWhdhez	桌面电源组帮助卷的繁体中文（通用）本地化版本。
SUNWhdicn	CDE 图标的繁体中文本地化版本。
SUNWhdim	CDE Imagetool 的繁体中文本地化版本。
SUNWhdwm	CDE 桌面窗口管理器的繁体中文本地化版本。
SUNWhepmw	电源管理 OW 公用程序的繁体中文 (EUC) 本地化版本。
SUNWhfdl	Adobe Postscript TCP/IP 打印机的繁体中文 Solaris 字体下载程序。
SUNWhj2p	Java Plug-in 1.2.2 的繁体中文本地化版本。
SUNWhj2rt	Java 虚拟机和核心类库（繁体中文增补）。
SUNWhjmfp	JMF 播放器的繁体中文本地化版本。
SUNWhkcsr	繁体中文 (EUC) KCMS 运行时环境。
SUNWhkdcl	Solaris 无盘客户机管理应用程序的繁体中文（香港）本地化版本。
SUNWhkeex	繁体中文（香港 BIG5）语言环境的专用文件。它是运行繁体中文（香港） BIG5）语言环境所必需的软件包（64 位）。

表 C–9 繁体中文 （续下）

软件包名称	描述
SUNWhkeue	该软件包包含繁体中文（香港）语言环境的专用文件。它是运行繁体中文（香港）语言环境所必需的软件包。
SUNWhklvmg	Solaris 卷管理应用程序的繁体中文（香港）本地化版本。
SUNWhkmga	繁体中文（香港）Solaris 管理应用程序。
SUNWhkrmui	繁体中文（香港）资源管理用户界面组件。
SUNWhksmc	繁体中文（香港）Solaris 管理控制台 2.0。
SUNWhkxe	繁体中文（香港）X Windows 平台软件包。
SUNWhleex	繁体中文语言环境的特定文件。它是运行繁体中文语言环境所必需的软件包（64 位）。
SUNWhleue	繁体中文语言环境的特定文件。它是运行繁体中文语言环境所必需的软件包。
SUNWhlvma	Solaris 卷管理 API 的繁体中文本地化版本。
SUNWhlvmg	Solaris 卷管理应用程序的繁体中文本地化版本。
SUNWhmga	繁体中文 Solaris 管理应用程序。
SUNWhorte	繁体中文 OPENLOOK 工具箱运行时环境软件包。
SUNWhos	该软件包包含繁体中文语言环境的专用文件。它是运行繁体中文语言环境所必需的软件包。
SUNWhpdas	用于同步桌面应用程序和 Palm Pilot PDA 的工具的繁体中文本地化版本。
SUNWhreg	用于 Solaris 用户注册的繁体中文本地化版本。
SUNWhrmui	繁体中文资源管理用户界面组件。
SUNWhsadl	Solstice Admintool 启动程序和相关库的繁体中文本地化版本。
SUNWhscgu	繁体中文可本地化的 Solaris 智能卡管理 - 图形用户界面组件。
SUNWhsfw	用于 SFW 统一的繁体中文可本地化信息文件。
SUNWhsmc	繁体中文 Solaris 管理控制台 2.0。
SUNWhtltk	繁体中文 ToolTalk 运行时软件包。
SUNWhttfe	繁体中文 True Type 可选字体软件包扩展。
SUNWhudc	Solaris CDE 环境的用户定义字符工具繁体中文 (EUC) 本地化版本。
SUNWhwbc	Solaris WBEM 服务的繁体中文本地化版本。
SUNWhwbcp	繁体中文 OpenWindows 二进制兼容性软件包。
SUNWhwdev	Solaris WBEM 服务的繁体中文本地化版本。
SUNWhwsr2	Solaris 产品注册的繁体中文本地化版本。

表 C–9 繁体中文 （续下）

软件包名称	描述
SUNWhwsrv	Solaris 产品注册查看器的繁体中文本地化版本。
SUNWhxe	繁体中文 X Windows 平台软件包。
SUNWhxman	繁体中文 X Windows 联机用户手册页软件包。

下表列出了共享软件包及其内容。

表 C–10 亚洲共享软件包

软件包名称	描述
SUNWabcp	SunOS 4.x 二进制兼容性的亚洲通用文件。
SUNWudct	Solaris CDE 环境的用户定义字符工具。

索引

数字和符号

@ls numerals=nominal:national，125
16 位 Unicode 3.0 代码集，133

A

API，43
AttrObject，125

C

catgets()，49
CDE, en_US.UTF-8 语言环境支持，92
CDE 输入法，94
Context，125
.cshrc，114
CSText，126
ctype 宏，42

D

DST（夏时制），28
Dt Apps，122
DtMail, MIME 字符集，116
dtterm，113

E

editing behavior，126
en_US.UTF-8
  fontset 定义，119
  支持，92

F

FontSet 定义，119

G

genmsg 公用程序，49
GMT 偏移，28

H

HKSCS iconv 模块，22

I

IA, 键盘，74
iconv, 日语字符代码转换，86
iconv 转换模块
  IBM/Microsoft EBCDIC，178
  IBM/Microsoft EBCDIC/PC 代码页，180
  PC 代码页，178
  与 Unicode 相关的，157
ISO-10646，92
ISO 拉丁-1，25
ISO8859-n 字符支持，93

L

LANG, 环境变量，112
LANG 环境变量，111
Latin-1 书写序列，100
Latin-1、Latin-2 或 KOI8-R 的终端支持，114
Latin-2 书写序列，101
Latin-3, 书写序列，104
Latin-4 书写序列，103
Latin-5 书写序列，104
Latin-9 书写序列，104
Latin-n 终端，114
layout behavior，126
layoutDirection，132
LayoutObject，124, 125
LC_ALL，25
libc，40, 42
libc
  查询语言环境数据，44
  代码转换，43
  多字节处理，47
  格式化宽字符输入和输出，48
  货币处理，46
  宽字符，44
  宽字符串，49
  宽字符和字符串处理，47
  宽字符输入和输出，49
  日期和时间格式，46
  信息传送，43
  修改和查询语言环境，44
  正规表达式，43
  字符分类和拼写，44
  字符排序，46
locale 公用程序，111

M

m_create_layout()，125
mbtwoc()，43
modinfo 命令，113
mp，22, 141, 142
mp，16
mystreams 文件，115

N

Numerals，133
NUMERALS_CONTEXTUAL，133
NUMERALS_NATIONAL，125
NUMERALS_NOMINAL，125

O

Orientation，125
OutToInp，127

P

prolog 文件，149
  打印布局，149
  通用，149
  依赖于语言环境，149
Property，127
pseudo-XOC，124

S

/sbin/sh 命令，41
setlocale 手册页，112
ShapeCharset，125, 133
Solaris 9
  朝鲜语本地化，87
  功能，22
  日语输入系统，85
  亚洲的语言环境，75
  语言环境内容，55, 75
  中文代码集，80
SPARC 键盘，66
strchg 命令，114
strconf 命令，114
STREAMS 模块
  32 位 STREAMS，112
  64 位 STREAMS，112
  TTY 环境设置，112
  u8lat1，114
  u8lat2，114
  保存设置，114
  装入，112

T

TextField, 132
TextShaping, 125
TTY 环境设置, 112
Tuthill, Bill, 18
TypeOfText, 125, 126

U

UIL, 132
Unicode 3.1, 支持, 92
Unicode 概述, 91
/usr/include/ctype.h, ctype 宏, 42
/usr/ucb/stty, 使用 /bin/stty 目录, 114
UTC, 28
UTF-8 encoding, 92

X

X/Open-Uniforum Joint Internationalization Working Group, 92
X 逻辑字体描述, 146
XLFD, 146
XmALIGNMENT_CENTER, 126
XmALIGNMENT_END, 126
XmCR_MOVING_INSERT_CURSOR, 128
XmDEFAULT_DIRECTION, 124
XmEDIT_LOGICAL, 127, 128, 137
XmEDIT_VISUAL, 127, 137
XmFont_IS_XO, 125
XmFONT_IS_XOC, 125, 135
XmLabel, 124, 133
XmLabelG, 124
XmList, 124
XmNalignment, 126, 132
XmNAlignment, 133
XmNeditPolicy, 127, 137
  left-character(), 128
XmNfont, 125
XmNfontName, 125
XmNfontType, 125
XmNgainPrimaryCallback, 128
XmNlabelString, 133
XmNlayoutAttrObject, 125
XmNlayoutDirection, 123, 124, 131, 132
XmNlayoutModifier, 124, 125, 132, 133
XmNlayoutModifier, 125
XmNmotionVerifyCallback, 128
XmNrenderTable, 126, 138
XmNrenditionTag, 126
XmRenderTableAddRenditions, 135
XmRendition, 123, 124, 125, 126, 132
XmRendition{Retrieve,Update}, 125
XmString, 124, 131
XmSTRING_COMPONENT_LAYOUT_PUSH, 124
XmSTRING_COMPONENT_LOCALE_TEXT, 124
XmSTRING_COMPONENT_TEXT, 124
XmSTRING_COMPONENT_WIDECHAR_TEXT, 124
XmStringDirection, 124, 131
XmStringDirectionCreate, 131
XmText, 124, 126, 132
XmTextField, 124, 126
XmTextFieldGetLayoutModifier, 130
XmTextFieldSetLayoutModifier, 131
XmTextGetLayoutModifier, 130
XmTextSetLayoutModifier, 131
XoJIG, 92
XPG4 应用程序, 42
xpr, 149

阿

阿拉伯文键盘, 105
阿拉伯语, 字符支持, 93

保

保存, STREAMS 模块设置, 114

本

本地化, 定义, 23

比

比利时文键盘, 106

波
波兰语, 字符支持，93

布
布局服务，125
布局修饰符定向，125

部
部分本地化版本软件包, Netscape Communicator 4.7，185
部分本地化软件包，185
  ISO8859，188
  澳大拉西亚，185
  朝鲜语，189
  东欧，186
  繁体中文，187
  南美，190
  南欧，190
  瑞典语，190
  泰语，190
  西欧，190
  印度语，188
  中美洲，186
  中文、日语和朝鲜语的公用文件，185

查
查表输入法模式，111

朝
朝鲜语
  `iconv` 模块，89
  `input method`，87
  `locales`，87
  本地化，87
  位图字体，88, 89
  字符支持，93
朝鲜语 TrueType，89
朝鲜语输入模式，108
朝鲜语文本
  朝鲜语汉字，33
朝鲜语文本 (续下)
  朝鲜语文字，33
  语言介绍，33
朝鲜语文字，33
朝鲜语语言软件包，201
朝鲜语中汉字，33

创
创建信息目录，49

打
打印布局 prolog 文件，149
打印过滤器，22, 141
打印过滤器增强，16
打印过滤器增强模式，142

代
代码集, 字符支持，93
代码集独立性 (CSI)
  Java 国际化，37
  Shift-JIS 代码集，37
  Solaris 9 环境库，39
  Solaris 9 环境命令，38
  动态链接应用程序，40
  多字节字符支持，40
  进程代码格式，40
  扩展UNIX 码 (EUC)，37
  文件代码编码，38
  语言环境数据库格式和结构，39
代码转换，115
代码转换 STREAMS 模块，112
代码转换模块，157, 178, 180

德
德语, 字符支持，93
德语语言软件包，195

动
动态链接，40

多
多字节文件代码转换，43

俄
俄语, 字符支持，93

法
法语语言软件包，193

繁
繁体中文
  input method，81
  locales，81
繁体中文（香港特别行政区）, UTF-8 语言环境，22
繁体中文（中国香港特别行政区）, Big5+HKSCS 语言环境，22
繁体中文输入模式，110
繁体中文语言软件包，206

复
复杂文本布局 (CTL)，16
  Motif，123
  Motif 库移植说明，138
  PLS，121
  XOC 资源，122
  编辑绘制，134
  布局方向，132
  创建绘制，133
  创建绘制表，135
  动态文本小部件，121
  段排序，121
  复杂语言成形，121
  技术概述，121
复杂文本布局 (CTL) (续下)
  键盘选定，138
  静态和动态文本，121
  连字，121
  鼠标选定，137
  水平制表符，136
  体系结构概述，122
  文本定向，121
  文本绘制，121
  文本资源和几何图形，138
  音调符号，121
  语言引擎，121
  制表，121
  字符成形，121
复杂文本布局(CTL), X 库输出模块 (XOM)，122

格
格林威治时间偏移，28

公
公用程序
  genmsg，49, 50
  locale，111

国
国际化
  ISO 拉丁-1，25
  定义，23
国际化 API，43
国际化应用程序，118

环
环境变量, LANG，112

绘
绘制表，135

简

简体和繁体中文
 辅助窗口支持，77
 输入法，76
简体中文
 `locales`，78
 输入法，79
 字符支持，93
简体中文输入模式，109
简体中文语言软件包，191

键

键盘，34, 62
 在 IA 上更改键盘，74
 在 SPARC 上更改，66
键盘布局
 4、5 和 5c 型键盘，64
 比利时语键盘，66
 丹麦语键盘，67
 德语键盘，68
 法语键盘，68
 繁体中文键盘，72
 芬兰语键盘，67
 荷兰（荷兰语）键盘，69
 美国/UNIX 键盘，73
 美国键盘，73
 挪威语键盘，70
 葡萄牙语键盘，70
 瑞典语键盘，71
 瑞士（德语）键盘，71
 瑞士（法语）键盘，71
 土耳其语 F 型键盘，72
 土耳其语 Q 型键盘，66
 西班牙语键盘，70
 意大利语键盘，68
 英国键盘，73

捷

捷克语, 字符支持，93

静

静态链接，40

开

开发者群，92

可

可移植布局服务 (PLS)，121
 绑扎，121
 对称交换，121
 数字成形，121
 双向性，121
 特征列表，121
 字符串验证，121

库

库, 将应用程序链接到，40

宽

宽字符, 进程代码，43
宽字符表达式，37

链

链接应用程序，40

命

命令, 具有 CSI 特征，38

欧

欧元，22

配

配置文件的本地化，143

日
日期格式，28
日语
  iconv 模块，22, 86
  locales，84
  本地化，84
  输入系统，85
  完全本地化软件包，87
  字符集，84
  字符支持，93
  字体，84
日语输入模式，108
日语文本
  片假名，32
  平假名，32
  日语汉字，32
  语言介绍，32
日语语言软件包，198
日语中的汉字，32
日语中的片假名，32
日语中的平假名，32

瑞
瑞典语语言软件包，205

设
设置
  TTY 环境，112
  终端选项，113

时
时区，28

输
输入模式
  Cyrillic，110
  en_US.UTF-8 语言环境，94
  西里尔文，104, 105, 106, 108, 109, 110, 111
  希腊文，105
  希腊语，109
输入模式 (续下)
  英语，96

书
书写键，34
书写键序列
  Latin-1，97
  Latin-3，103
书写序列，105, 106, 108, 109
  Latin-1，100
  Latin-2，101
  Latin-3，104
  Latin-4，103
  Latin-5，104
  Latin-9，104

数
数字格式
  分隔符，29
  惯例，29
  基数字符，29
  千位分隔符，29
  小数点位置，29

斯
斯堪的纳维亚和波罗的海语, 字符支持，93

泰
泰语
  输入法，78
  字符序列检查，77
  字符支持，93
泰语 ISO8859–1，22
泰语 UTF-8，22
泰语键盘，110
泰语文本，33

通
通用 prolog 文件，149

土
土耳其语, 字符支持，93
土耳其语 F 型键盘，66

完
完全 Solaris 语言环境，26
完全本地化版本软件包
    朝鲜语，201
    共享，209
    瑞典语，205
    西班牙语，203
完全本地化软件包
    德语，195
    法语，193
    繁体中文，206
    简体中文，191
    日语，87, 198
    意大利语，196

西
西班牙语
    语言软件包，203
    字符支持，93
西里尔文输入模式，105, 106, 109, 110, 111
西欧语言
    字符支持，93
西欧字母表，32

希
希伯来文键盘，107
希伯来语, 字符支持，93
希伯语文本，34
希腊文, 输入模式，105
希腊文 UNIX 键盘，106
希腊文欧元键盘，106
希腊文输入模式，105
希腊语
    输入模式，109
    字符支持，93

系
系统库, 将应用程序链接到，40

夏
夏时制 (DST)，28

协
协调通用时间，28

新
新键盘支持，16

信
信息目录，49

匈
匈牙利语, 字符支持，93

亚
亚洲 Solaris 的排序语言环境，22
亚洲受支持的语言环境，76

依
依赖于语言环境的 prolog 文件，149

意

意大利语语言软件包，196

印

印地文, 字符支持，93
印地语文本，34
印地语语言支持，22

英

英语
  输入模式，96
  字符支持，93

应

应用程序
  FontSet/XmFontList 定义，118
  XPG4，42
  国际化，118
  链接到系统库，40

映

映射到字体组，144

语

语言环境，37
  代码集，25
  单词分隔符，32
  定义，25
  环境变量，111, 112
  货币，23
  货币格式，30
  基本语言，25
  键盘差异，34
  排序顺序，32
  日期格式，28
  时间格式，27
  使用国家/地区，25

语言环境 (续下)
  数字格式，29
  文化惯例，26
  页大小，34
  种类，27
  字符集，32
语言环境环境变量，111
语言环境内容，55, 75
语言选择，94

中

中国，33
中文文本, 语言介绍，33
中文中的汉字，33
中文中的拼音，33

终

终端
  Latin-n，114
  Latin-n 终端，114
终端选项, 设置，113

转

转换，115
  多字节和宽字符进程代码，43

装

装入, STREAMS 模块，112

字

字符串
  `XmTextFieldGetLayoutModifier`，130
字符支持，93
字母表，32
字体别名，144
字体组定义，144